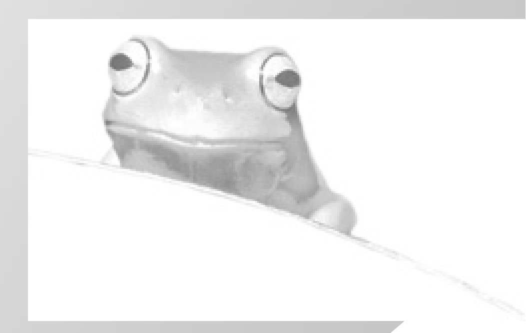


DATENBANKEN

in der Sekundarstufe II

Theorie und Praxis

Version 1.000001

J. Burkert
R. Lächa
D. M. Meyer [1]



[1] Die Darstellungen stellen eine Erweiterung des hervorragenden Werkes "Jürgen Burkert, Roland Lächa, Hessisches Institut für Bildungsplanung und Schulentwicklung (HIBS), Wiesbaden" dar.

Mein Dank gilt ferner den vielen Autoren mit offener Einstellung, die ihre Materialien frei im Internet zur Verfügung stellen und die viele der hier aufgeführten Darstellungen inspiriert haben! Sollte sich ein Kollege bzgl. des Copyrights verletzt fühlen oder möchte namentlich genannt werden, möge er sich einfach kurz an mich wenden.

Inhaltsverzeichnis

Unterrichtseinheit “Datenbanken“

Hinweise zu diesen Unterlagen

Vorliegende Unterlagen dienen als Begleitskript zum Unterricht. Sie verfolgen mehrere Primärziele:

- Es soll möglich sein Stoff selbständig Vor- und Nachzubereiten (permanente Hausaufgabe).
- Ein Selbststudium soll ermöglicht und motiviert werden.
- Im Unterricht erarbeitete Konzepte sollen hier – in ausführlicher Darstellung - nochmals nachgelesen werden können.
- Themata des Unterrichts werden an Hand weiterer Beispiele vertieft.
- Eine Teilentlastung bzgl. des Mitschreibens soll erreicht werden.
- Vernünftige mehrstufige Lernarbeit - bestehend aus der Unterrichtsvorbereitung, dem Unterricht selbst, den Hausaufgaben, der Nachbereitung, der Reflexion und der Selbstlernphasen wie Klausurvorbereitung, in Erinnerung rufen, schreiben von Zusammenfassung uvm. - soll unterstützt und gefördert werden.

Im Unterricht werden die relevanten Konzepte erarbeitet. Diese Unterlagen reflektieren dies – teilweise in etwas vertiefter Form – und bieten die Möglichkeit sowohl das eigene Verständnis zu prüfen als auch die Inhalte nochmals textuell und anders dargestellt vor sich zu haben.

Zur schnellen Auffindung werden folgende Symbole benutzt:

	Arbeitsauftrag	Eine zu bearbeitende Übungsaufgabe
	Wichtige Information	Eine besonders wichtige Information, z.B. ein mathematischer Satz oder ein zentrales Prinzip. Diese Dinge müssen unbedingt beherrscht werden und im Zusammenhang erklärt werden können. Sie eignen sich besonders zur Erstellung einer Zusammenfassung.
	Zusammenfassung	Am Ende eines sinnvollen Abschnittes, meist eines Kapitels, gibt es Zusammenfassungen, die die wichtigsten Punkte zusammenfassen. → Kandidaten für eigene Überblickzusammenfassungen!
	Ausblicke	Ausflüge in unentdeckte Lande, Exkurse, die für den Unterricht nicht zwingend notwendig sind
	Projekte	Projektaufträge in Einzel- oder Gruppenarbeit.

	Zusatzinformationen	Weiterführende Hinweise, Exkurse oder Zusatzinformationen, die zum tieferen Verständnis beitragen, historische Daten liefern oder neuere Entwicklungen aufzeigen. Sie sind für das Primärverständnis des Stoffes im ersten Anlauf nicht so wesentlich.

Neben dem Text finden sich wesentliche Stichpunkte zum schnellen Auffinden als Einzelwort oder in kurzen Sätzen.

Arbeitshinweise und Lerntechniken

Für das Erlernen der wichtigsten Konzepte der Informatik sind die gleichen Aspekte zu Berücksichtigen, die auch in anderen Fächern gelten. Eine seriöse Lernarbeit ist Voraussetzung für eine erfolgreiche Bewältigung des Inhaltes. Zwar lernt jeder anders, jedoch gibt es einige Lerntechniken, die unbedingt empfehlenswert sind. Dazu gehören vor allem folgende Dinge:

- Folgen Sie unbedingt Aufmerksam dem Unterricht und beteiligen Sie sich aktiv.
- Stellen sie offen Fragen und melden Sie sich rechzeitig, bevor Sie unter Umständen „abgehängt“ werden. Das würde Ihre Arbeit nur um ein Vielfaches sinnlos vermehren.
- Seien Sie unbedingt ehrlich zu sich selbst. Die Selbsteinschätzung ist eine zentrale Fähigkeit, die es zu entwickeln gilt.
- Lernen Sie permanent! Einen Tag vor der Klausur hilft eh nichts mehr. Und selbst wenn. Sie lernen für sich, nicht für andere. Nur langfristige Arbeit bleibt auch im Kopf erhalten!
- Motivieren Sie sich selbst. Ohne Spaß wird alles nur viel zäher.
- **Schreiben Sie sich unbedingt Zusammenfassungen**. Der Inhalt ist dabei auf Ihr persönliches Verständnis abgestimmt.
- Schreiben Sie sich in regelmäßigen Abständen Zusammenfassungen der Zusammenfassungen, auch “Spickzettel“ genannt. Sie enthalten ganz prägnant und komprimiert das für Sie wesentliche.
- **Bearbeiten Sie unbedingt selbständig die Übungsaufgaben** und notieren Sie sich ausführlichst Ihre Lösungen. Nehmen Sie dabei vor allem Bezug zur Theorie, indem Sie z.B. die angewandten Kriterien bei der Normalisierung als solche neben den konkreten Normalisierungsschritten notieren. So verknüpft sich das Wissen und Sie können auch morgen noch Ihre Lösungen nachvollziehen. **Ferner erklären Sie sich Ihre Lösung durch einen sorgfältigen Lösungsweg praktisch selbst**.

Es schadet nicht, die bereits bewältigten Übungsaufgaben im Rahmen der Klausurvorbereitung nochmals durchzurechnen.

1 Datenbanken

1.1 Datenbanken im Informatik-Unterricht

Datenbanken besitzen in der Praxis eine immense Bedeutung. Sie bilden die Basis zahlreicher Informationssysteme, ob im Personalwesen, im CAD (computer aided design) oder in modernen Verzeichnisdiensten wie dem active directory, Datenbanken bilden den Kern vieler Informationssysteme.

Dabei kommt der

- *Modellbildung*

und der

- *Abstraktion*

besondere Bedeutung zu. Ein Ausschnitt der Wirklichkeit wird adäquat durch ein Modell erfasst, welches angemessen Detailliert sein muss, d.h. alle für das Ziel relevanten Aspekte erfassen muss, sich nicht jedoch in Details verlieren darf. Dies ist eine Fertigkeit, die in vielen Bereichen des Lebens gefordert ist, auch im nichttechnischen Bereich.

Zudem sind Datenbanken ein guter Repräsentant vieler Problemstellungen in der Informatik. Datenstrukturen, Wertebereiche, Modellbildung, Parametrisierung, Datenkapselung, Datenorganisiation, Programmierung, Mensch-Maschine-Kommunikation uvm. finden sich hier wieder.

Entsprechend sinnvoll ist der Einsatz von Datenbanken im Informatik-Unterricht. Dabei wird exemplarisch das RDBMS[1] Access verwendet. Wichtiger sind jedoch die theoretischen Grundlagen, da diese Bestand haben und nicht so schnell veralten. Access dient nur der exemplarischen Umsetzung, wird evtl. mit MySQL weitergeführt und kann als ein Beispiel von vielen dienen.

Das Bildungsziel ist dabei neben konkreten Fähigkeiten im Umgang mit Datenbanksystemen die "**Analyse, Beschreibung und Modellierung komplexer Systeme**".

1.2 Inhalt der Unterlagen

Wir werden Theorie und Praxis von Datenbanken behandelt und an geeigneten Beispielen erläutert. Dabei sollte erkannt werden, dass Datenbanken vor allem Modellierungsergebnisse sind, wobei die Modellierungsprinzipien denen der OOM/OOP sehr stark ähneln.

Als Ausgangspunkt dienen dabei verschiedene kleine Probleme, aber auch zwei durchgängige größere Projekte, die "Informatik-Bücherei" und eine Kursverwaltung. Selbige Projekte sollten anschaulich und einleuchtend sein, zumal an der Schule jeder mit diesen Dingen zu tun hat. Entsprechend sinnvoll wäre es auch, die Lösungsvorschläge zu erweitern, so dass sie in der Realität als Software eingesetzt werden könnten. Dazu sei ausdrücklich angeregt!

[1] Relationales Datenbank Management System

1.3 Wir lernen eine Datenbank kennen

Bevor wir uns mit der Theorie von Datenbanken systematisch auseinander setzen, wollen wir zunächst einmal eine fertige Datenbank erkunden. Dadurch entsteht ein Bild dessen, woraus eine Datenbank besteht und was sie leisten soll. Ferner können Begriffe der Theorie dann direkt mit bekannten Umsetzungen in Verbindung gebracht werden.

Aufgabe 1

ACHTUNG: Speichere all deine Daten immer ausschließlich in deinem privaten Verzeichnis



Schreibe die Lösung jeder Aufgabe mit blauer Farbe direkt unter jede Aufgabe.

Öffne die Beispieldatenbank “Nordwind” mit MS Access. (Access starten und “Nordwind” auswählen.) Öffne das Formular “Bestellungen”.

1. Welche Informationen werden hier angezeigt? Für welchen Personenkreis könnte dieses Formular gedacht sein?
2. Welche Informationen können verändert werden?
3. Sieh Dir verschiedene Bestellungen an. Wie kommst Du zur 1. bzw. zur letzten der Bestellungen?
4. Wie kannst Du eine neue Bestellung eingeben? Gib folgende Bestellung ein:

   Rechnung an “The Cracker Box”, Versand an Dich Bestelldatum 1.10.2001 Lieferdatum 8.10.2001

   1 x Camembert Pierrot

   2 x Mozzarella di Giovanni

   4 x Gustaf's Knäckebröd

   1 x Queso Manchego La Pastora

   3 x Scottish Longbreads (20% Rabatt, da diese Woche im Angebot)

   Welche Rechnungssumme erhältst Du?

Aufgabe 2

Vorsicht: Wenn Du an einer bestehenden Bestellung ein Feld veränderst, wird es beim Verlassen des Feldes automatisch gespeichert (ohne Nachfrage). So veränderst Du die ursprünglichen Daten unwiederbringlich.

1. Klicke in ein editierbares Feld und drücke dann die rechte Maustaste. Welche Funktionen kannst Du ausführen?
2. Beschreibe die Wirkung jeder Funktion kurz.
3. Wie viele Bestellungen hat der Verkaufsberater “Robert King” erzielt? (Gegenprobe: Wie viele Bestellungen sind nicht von “Robert King”?
4. Es sollen nur die Bestellungen angezeigt werden, die vom Verkaufsberater “Robert King” sind und “Ernst Handel” verrechnet wurden.
5. Sortiere alle Bestellungen nach dem Bestelldatum. Wann war die erste, wann die letzte Bestellung?

Aufgabe 3

1. Es wird ein neuer Kunde gewonnen:
   Michaeli Gymnasium, Hachinger Bach Str. 25, 81673 München
   Füge diesen neuen Kunden zu den schon bestehenden Kunden hinzu.
2. Nimm für diesen Kunden eine Bestellung auf.

Aufgabe 4

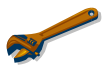
1. Wofür kann ein Datenbanksystem eingesetzt werden?
2. Welche Aufgaben gibt es bei der Erstellung und der Pflege einer Datenbank?
3. Wofür werden Formulare verwendet?
4. Wo stehen die Daten, sprich in welchen Strukturen werden sie von der Datenbank gespeichert?

Lösungsheft

Kapitel 1
–
Datenbanken im Informatikunterricht

Aufgabe 1

ACHTUNG: Speichere all deine Daten immer ausschließlich in deinem privaten Verzeichnis



Schreibe die Lösung jeder Aufgabe mit blauer Farbe direkt unter jede Aufgabe.

Öffne die Beispieldatenbank “Nordwind” mit MS Access. (Access starten und “Nordwind” auswählen.) Öffne das Formular “Bestellungen”.

1. Welche Informationen werden hier angezeigt? Für welchen Personenkreis könnte dieses Formular gedacht sein?
2. Welche Informationen können verändert werden?
3. Sieh Dir verschiedene Bestellungen an. Wie kommst Du zur 1. bzw. zur letzten der Bestellungen?
4. Wie kannst Du eine neue Bestellung eingeben? Gib folgende Bestellung ein:

   Rechnung an “The Cracker Box”, Versand an Dich Bestelldatum 1.10.2001 Lieferdatum 8.10.2001

   1 x Camembert Pierrot

   2 x Mozzarella di Giovanni

   4 x Gustaf's Knäckebröd

   1 x Queso Manchego La Pastora

   3 x Scottish Longbreads (20% Rabatt, da diese Woche im Angebot)

   Welche Rechnungssumme erhältst Du?

Aufgabe 2

Vorsicht: Wenn Du an einer bestehenden Bestellung ein Feld veränderst, wird es beim Verlassen des Feldes automatisch gespeichert (ohne Nachfrage). So veränderst Du die ursprünglichen Daten unwiederbringlich.

1. Klicke in ein editierbares Feld und drücke dann die rechte Maustaste. Welche Funktionen kannst Du ausführen?
2. Beschreibe die Wirkung jeder Funktion kurz.
3. Wie viele Bestellungen hat der Verkaufsberater “Robert King” erzielt? (Gegenprobe: Wie viele Bestellungen sind nicht von “Robert King”?
4. Es sollen nur die Bestellungen angezeigt werden, die vom Verkaufsberater “Robert King” sind und “Ernst Handel” verrechnet wurden.
5. Sortiere alle Bestellungen nach dem Bestelldatum. Wann war die erste, wann die letzte Bestellung?

Aufgabe 3

1. Es wird ein neuer Kunde gewonnen: Michaeli Gymnasium, Hachinger Bach Str. 25, 81673 München Füge diesen neuen Kunden zu den schon bestehenden Kunden hinzu.
2. Nimm für diesen Kunden eine Bestellung auf.

Aufgabe 4

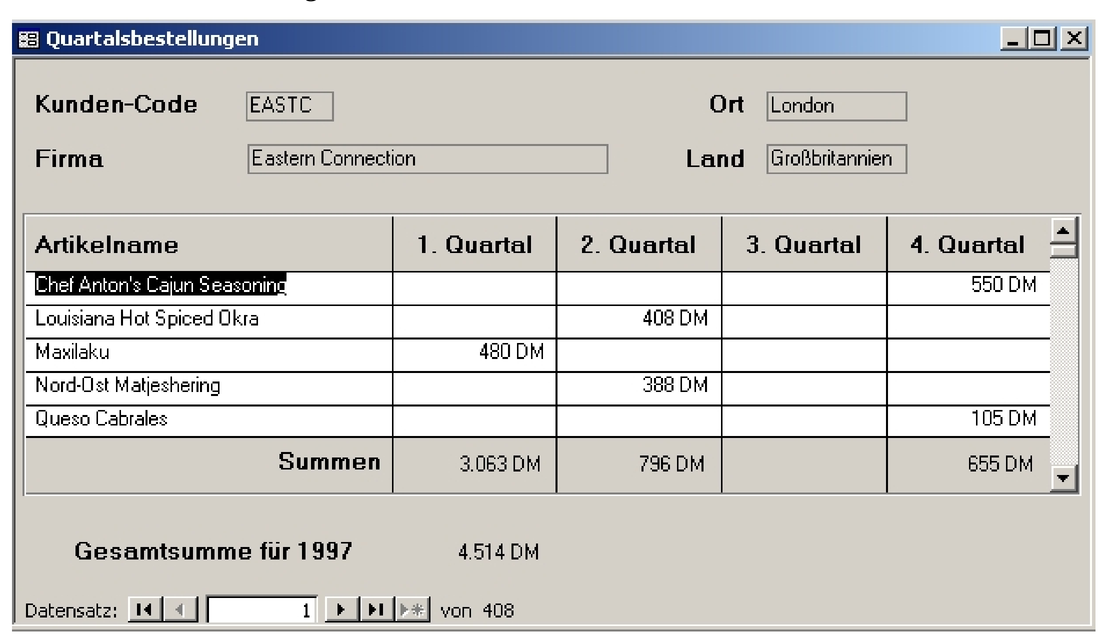
1. Wofür kann ein Datenbanksystem eingesetzt werden?
2. Welche Aufgaben gibt es bei der Erstellung und der Pflege einer Datenbank?
3. Wofür werden Formulare verwendet?
4. Wo stehen die Daten, sprich in welchen Strukturen werden sie von der Datenbank gespeichert?

Lösungen

Aufgabe 1

Öffne die Beispieldatenbank “Nordwind” mit MS Access. (Access starten und “Nordwind” aus dem Menüeintrag “?“ auswählen.) Öffne das Formular “Quartalsbestellungen”.

1. Welche Informationen werden hier angezeigt? Für welchen Personenkreis könnte dieses Formular gedacht sein?



Ziel ist wohl eine Übersicht über die #Bestellungen eines Kunden zu geben, z.B. um Fragen nach höherem Rabatt bewerten zu können.

2. Welche Informationen können verändert werden?

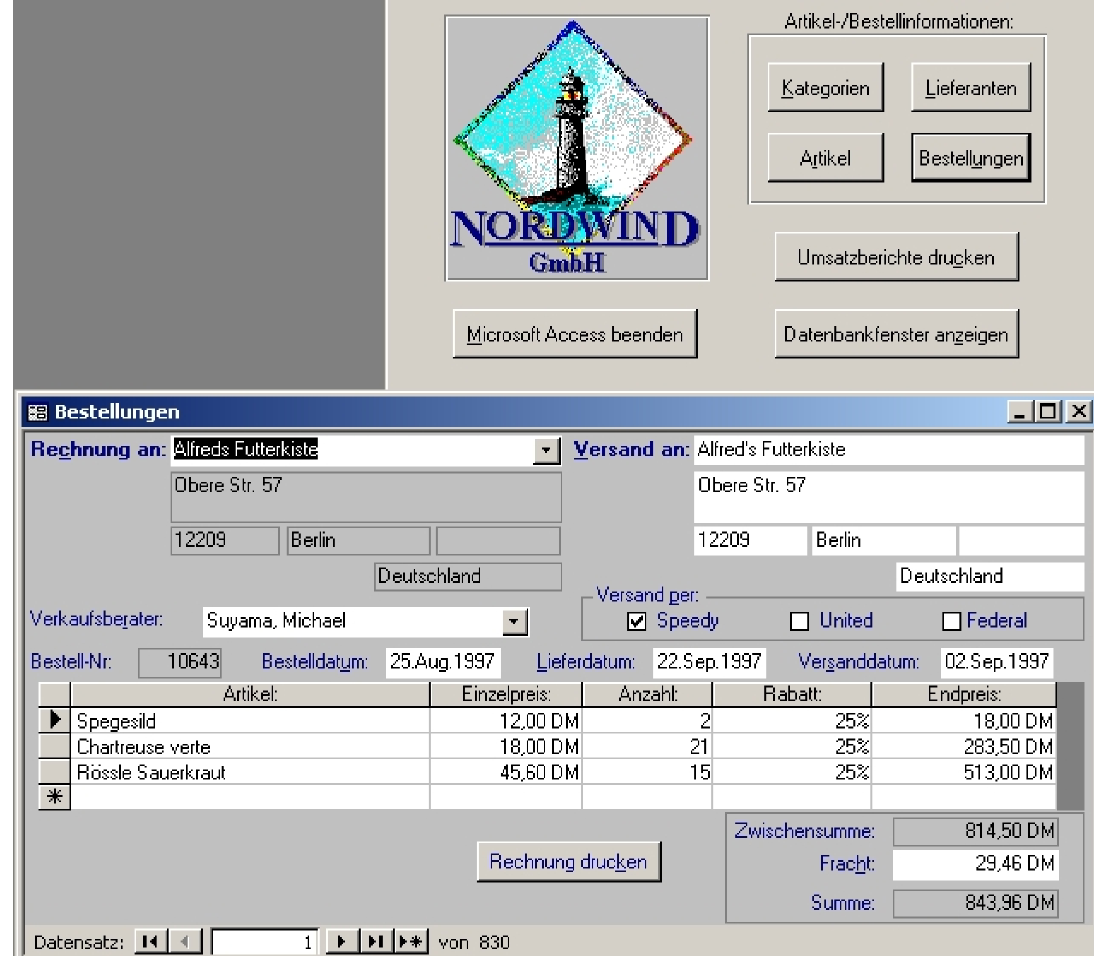
Keine, man kann lediglich von Datensatz zu Datensatz navigieren (read-only)

3. Sieh Dir verschiedene Bestellungen an. Wie kommst Du zur 1. bzw. zur letzten der Bestellungen?

Mit Pfeil ganz links/rechts |← und →|

4. Wie kannst Du eine neue Bestellung eingeben? Gib folgende Bestellung ein:

In dem aktuellen Formular gar nicht, ist ja read-only. Statt dessen gibt es ja das Hauptformular → Bestellungen:



Rechnung an "The Cracker Box", Versand an Dich
Bestelldatum 1.10.2001 Lieferdatum 8.10.2001

1 x Camembert Pierrot

2 x Mozzarella di Giovanni

4 x Gustaf's Knäckebröd

1 x Queso Manchego La Pastora

3 x Scottish Longbreads (20% Rabatt, da diese Woche im Angebot)

Welche Rechnungssumme erhältst Du?

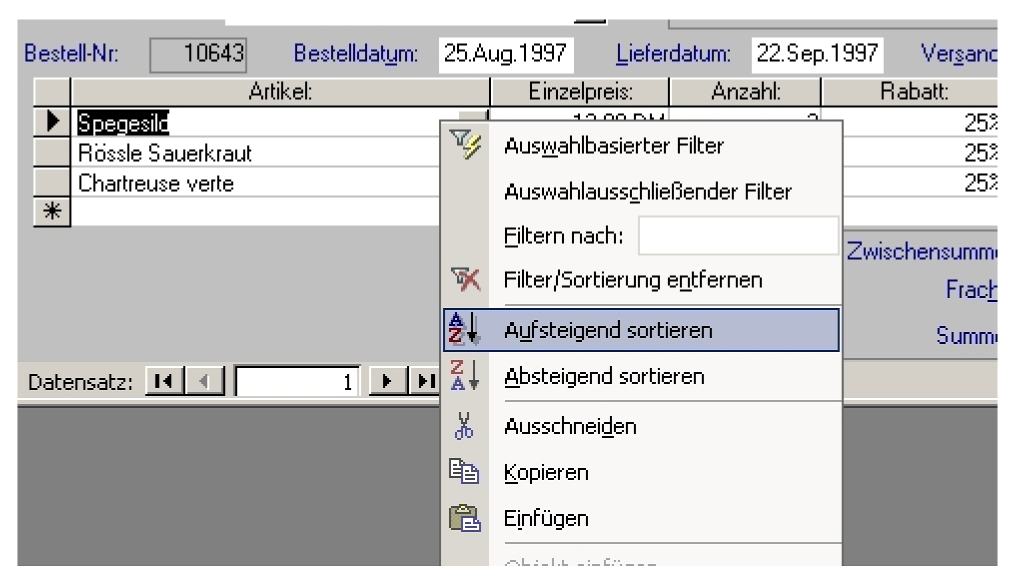
1. "The Cracker Box" unter "Rechnung an" aufrufen, "Versand an" mit deiner Adresse versehen, Artikeldaten eintragen

Aufgabe 2

Vorsicht: Wenn Du an einer bestehenden Bestellung ein Feld veränderst, wird es beim Verlassen des Feldes automatisch gespeichert (ohne Nachfrage). So veränderst Du die ursprünglichen Daten unwiederbringlich.

1. Klicke in ein editierbares Feld und drücke dann die rechte Maustaste. Welche Funktionen kannst Du ausführen?

Man kann Filter anwenden und sortieren lassen



2. Beschreibe die Wirkung jeder Funktion kurz.

Auswahlbasierter Filter: nur dieser Artikel wird angezeigt (mehrfach wenn mehrere vorhanden sind)

Auswahlausschließender Filter: genau anders rum

Filtern nach: Filterkriterium kann eingegeben werden, z.B. “<20“ bei Anzahl

Rest spricht für sich selbst

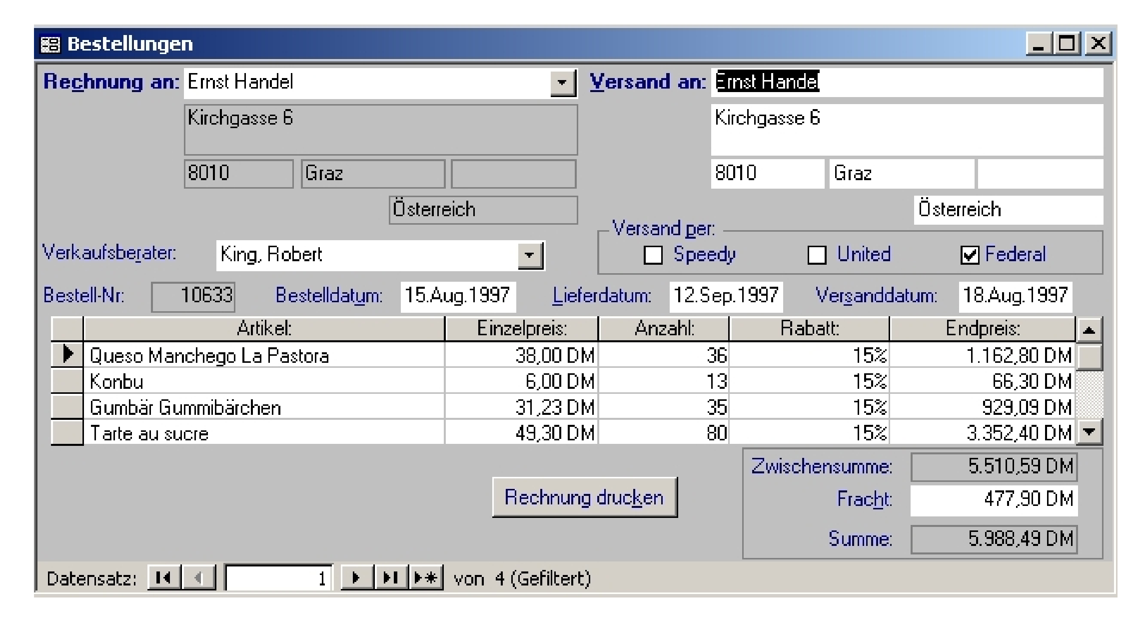
3. Wie viele Bestellungen hat der Verkaufsberater “Robert King” erzielt? (Gegenprobe: Wie viele Bestellungen sind nicht von “Robert King”?

72: rechte Maustaste auf das Feld “Verkaufsberater“, dann “Auswahlbasierter Filter“ wählen

830-72 sind nicht von ihm (rechte Maustaste: Filter aufheben)

4. Es sollen nur die Bestellungen angezeigt werden, die vom Verkaufsberater “Robert King” sind und “Ernst Handel” verrechnet wurden.

4: zuerst “Robert King“ wie bei 3 auswählen, dann “Filtern nach“ bei “Versand an“: Ernst*



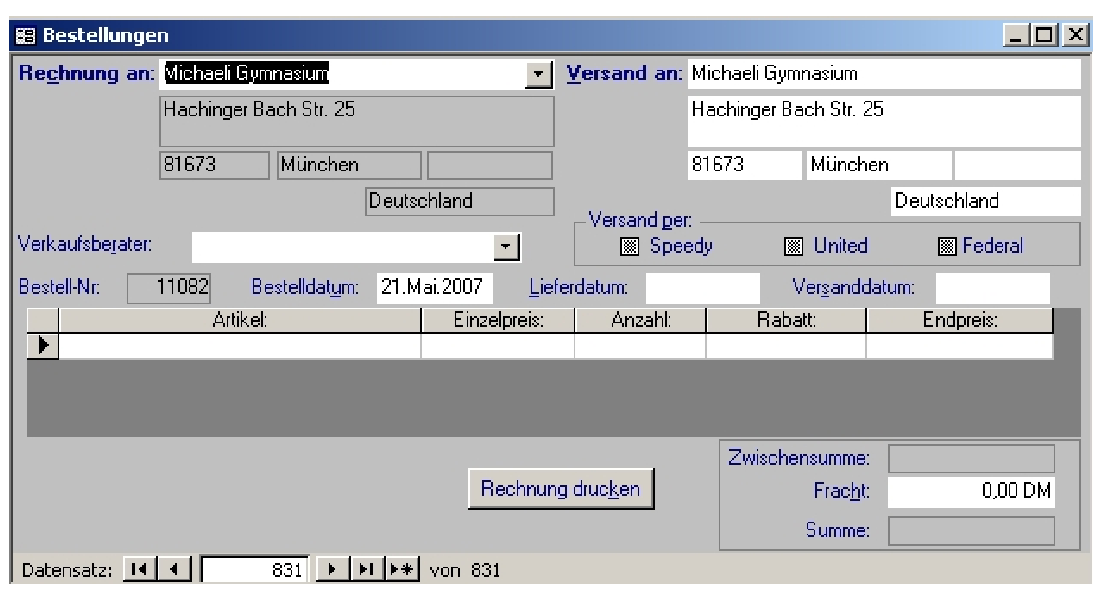
5. Sortiere alle Bestellungen nach dem Bestelldatum. Wann war die erste, wann die letzte Bestellung?

04.Juli 1996 - 06.Mai 1998 (rechte Maustaste auf “Bestelldatum“→aufsteigend Sortieren)

Aufgabe 3

1. Es wird ein neuer Kunde gewonnen: Michaeli Gymnasium, Hachinger Bach Str. 25, 81673 München Füge diesen neuen Kunden zu den schon bestehenden Kunden hinzu.
2. Nimm für diesen Kunden eine Bestellung auf.

Das geht nicht über ein Formular. Man muss die Daten direkt in der Tabelle “Kunden“ eintragen. Dann kann man wie gewohnt unter“Bestellungen“ die Michaeli Gymnasium auswählen und die Bestellung eintragen.



Aufgabe 4

5. Wofür kann ein Datenbanksystem eingesetzt werden?

Für alles, wo Daten verwaltet werden müssen. Z.B. bei Ärzten, in Reisebüros, …

6. Welche Aufgaben gibt es bei der Erstellung und der Pflege einer Datenbank?

Fragen,

- WAS die DB leisten soll
- WIE sie das machen soll (internes Modell und GUI)
- Pflege: Backup, neue Formulare, neue Abfragen, …

7. Wofür werden Formulare verwendet?

- zur Daten-I/O in gesicherter Form

- Realisierung des externen Modells, der Sichten von Anwendern auf die DB

8. Wo stehen die Daten, sprich in welchen Strukturen werden sie von der Datenbank gespeichert?

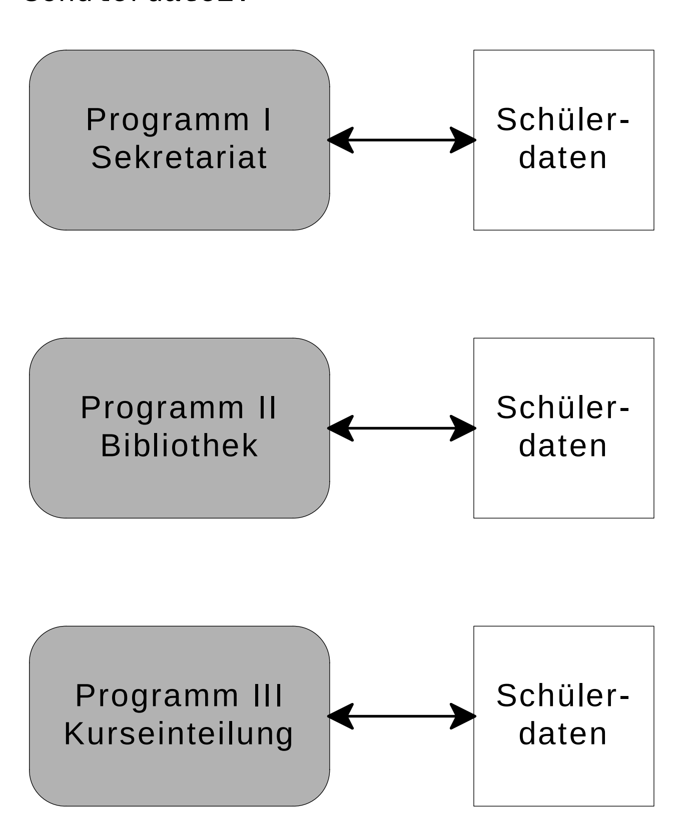
In Tabellen

2 Das konzeptuelle Modell

2.1 Datenbanken – Mehr als Tabellen? Wozu das Ganze?

Die Begriffe „Datenbank" (DB) und „Datenbankmanagementsystem" (DBMS) kann man sich vielleicht am besten verdeutlichen, wenn man sich ein Verfahren zur Verwaltung von Daten ansieht, das sich in den 60er und 70er Jahren etablierte und z.T. heute noch verwendet wird. Kennzeichnend dafür ist eine statische Zuordnung von Programmen und den Daten, die sie benötigen.

Beispiel*:* An der Schule werden an verschiedenen Stellen Stammdaten der Schüler wie Vorname, Nachname, Geburtsdatum benötigt, z.B. im Sekretariat, in der Bibliothek und beim Studienleiter, der die Kurse einteilt. Alle nutzen unterschiedliche Programme und verwalten dabei ihre eigene Schülerdatei.



Aufgabe:

a) Welche Nachteile hat eine Datenerfassung dieser Art?



b) Nennen Sie andere Bereiche, in denen Datenbanken eingesetzt werden.

Diese Probleme sollen mit dem Konzept der Datenbank und des DBMS vermieden werden. Das DBMS verwaltet zentral alle Daten eines Unternehmens, einer Institution, eines Projekts usw. Redundanzen und Inkonsistenzen können so vermieden werden, unterschiedliche Standards entfallen.
Die **Daten sind gemäß ihren logischen Zusammenhängen organisiert** und nicht entsprechend den Anforderungen spezieller Anwendungen. Das DBMS stellt Benutzern - das können Menschen, aber auch Programme sein - die Daten in einer für sie brauchbaren Form zur Verfügung. Ändert sich eine Anwendung, dann muss nicht mehr die Datenorganisation geändert werden, sondern lediglich eine neue „externe Sicht“ (View) auf die Daten geschaffen werden.

Auch Datensicherheit und Datenschutz können leichter gewährleistet werden. So erhält ein Benutzer durch seine spezifische Sicht nur Zugriff auf Daten, für die er eine Zugriffsberechtigung besitzt. Das DBMS kann durch spezifische Kontrollen beim Eingeben, Löschen und Ändern von Daten Fehler vermeiden helfen. So ist es z.B. nicht möglich, einem nicht existierenden Schüler Kurse zuzuweisen oder Kurse zu belegen, deren Kursnummer in der Datenbank nicht vorkommt. Man spricht hier von der Erhaltung der Datenintegrität.

Schön, warum aber organisiert man seine Daten nicht einfach mit einer Tabellenkalkulation - z.B. Excel -, die ja heute auch schon sehr mächtig sind und z.B. komfortable Suchfunktionen bieten?

<u>Aufgabe:</u>

a) Wie würde eine „Datenbank“ mit einer Tabellenkalkulation aussehen? Machen Sie sich das am besten an Hand eines konkreten Beispiels klar. Um den Überblick über die Ausgaben jedes Klassenmitglieds zu behalten, beschließen Sie, die Einkaufdaten in einer „Excel-Datenbank“ zu verwalten. Überlegen Sie sich, welche Daten Sie benötigen.

b) Öffnen Sie nun die Excel-Datei “*Kassenbon.xls*“ und schauen Sie sich die dortigen Vorschläge an. Vergleichen Sie Ihre Ideen mit denen in der Datei, sie müssen nicht schlechter sein!

c) Nehmen Sie dann einige Einfüge-/Lösch- und Suchoperationen vor. Was fällt Ihnen auf?

d) Welche Nachteile gibt es bei der Datenverwaltung mit einem Tabellenkalkulationsprogramm? Belegen Sie Ihre Vorschläge an Hand des konkreten Beispiels.

2.2 Die Ebenen eines Datenbankmanagement-Systems

Wir haben uns klar gemacht, dass es nicht das Wahre ist, umfangreiche Daten mit einem Tabellenkalkulationsprogramm zu verwalten. Deswegen heißt es ja auch schon Tabellenkalkulationsprogramm und nicht Datenbank. Doch momentan wissen wir eigentlich nur, was für Nachteile wir entdeckt haben, nicht wie wir sie beheben. Jedoch steht zu vermuten, dass aus den Nachteilen der Datenverwaltung per Tabellenkalkulation die Vorteile einer Datenbank erwachsen.

Bis dato haben wir uns gefragt, warum man nicht einfach alle Daten in eine große Tabelle packt und damit die Architektur einer Datenbank sozusagen von unten her betrachtet. Andererseits wird man einräumen müssen, dass derlei Überlegungen für den späteren Endanwender einer Datenbank unerheblich sind. Für ihn ist eher wichtig, wie die Benutzerschnittstelle ausgelegt ist, sprich wie “seine Sicht“ auf die Daten(bank) beschaffen ist.

Aus diesen ganz unterschiedlichen Betrachtungs- und Problembereichen ist folgende Unterteilung einer Datenbankarchitektur erwachsen. Man unterscheidet sinnvoll drei Ebenen:

Externe Sicht:

So stellt sich die Datenbank dem Benutzer dar.

Interne Strukturen der DB sind für den Anwender uninteressant. Er sollte die DB als Informationsspeicher bzw. als Informationslieferant wahrnehmen, die ihm genau das präsentiert, was er für sein jeweiliges Tätigkeitsfeld benötigt. So kann etwa die Lagerverwaltung einen ganz anderen Ausschnitt aus der Gesamtdatenbank zu sehen bekommen als z.B. die Personalverwaltung. Dies sind die entsprechenden externen Sichten auf die Datenbank.

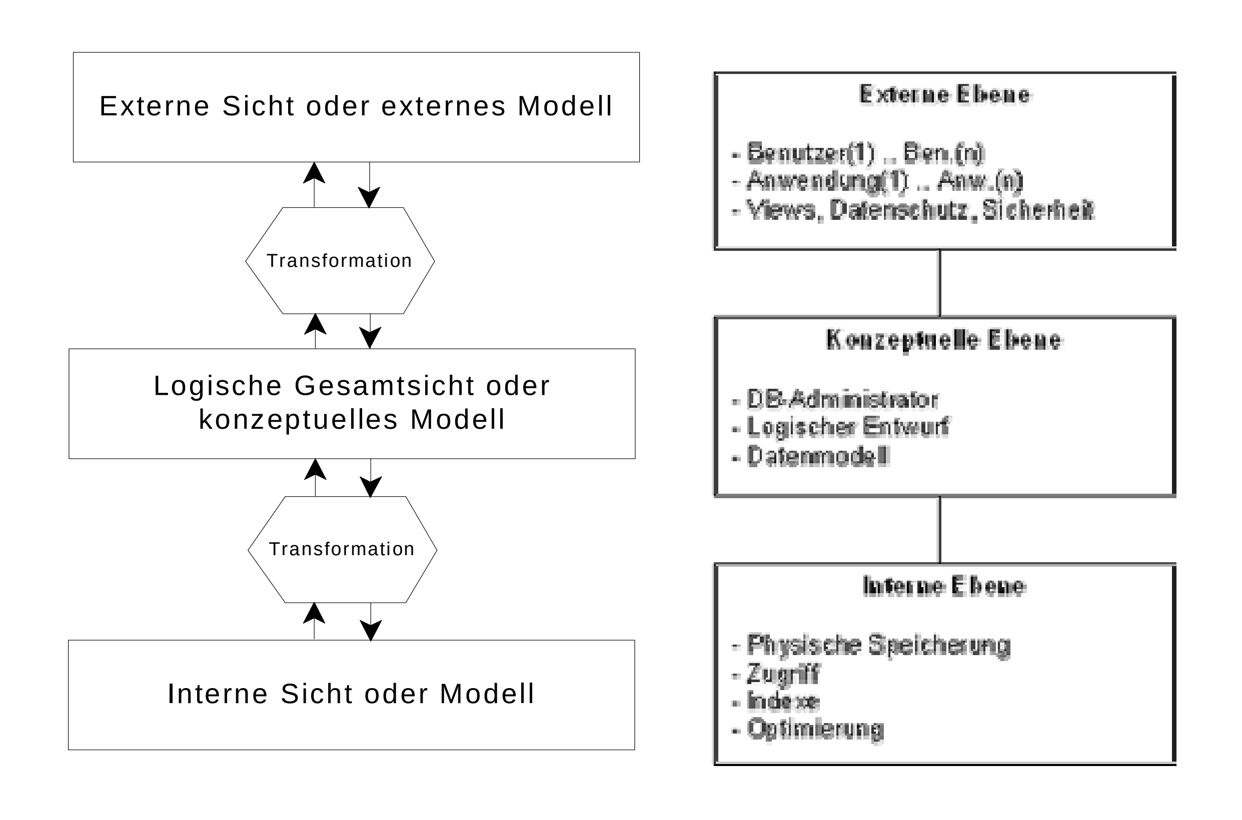
Logische/konzeptuelle Sicht:

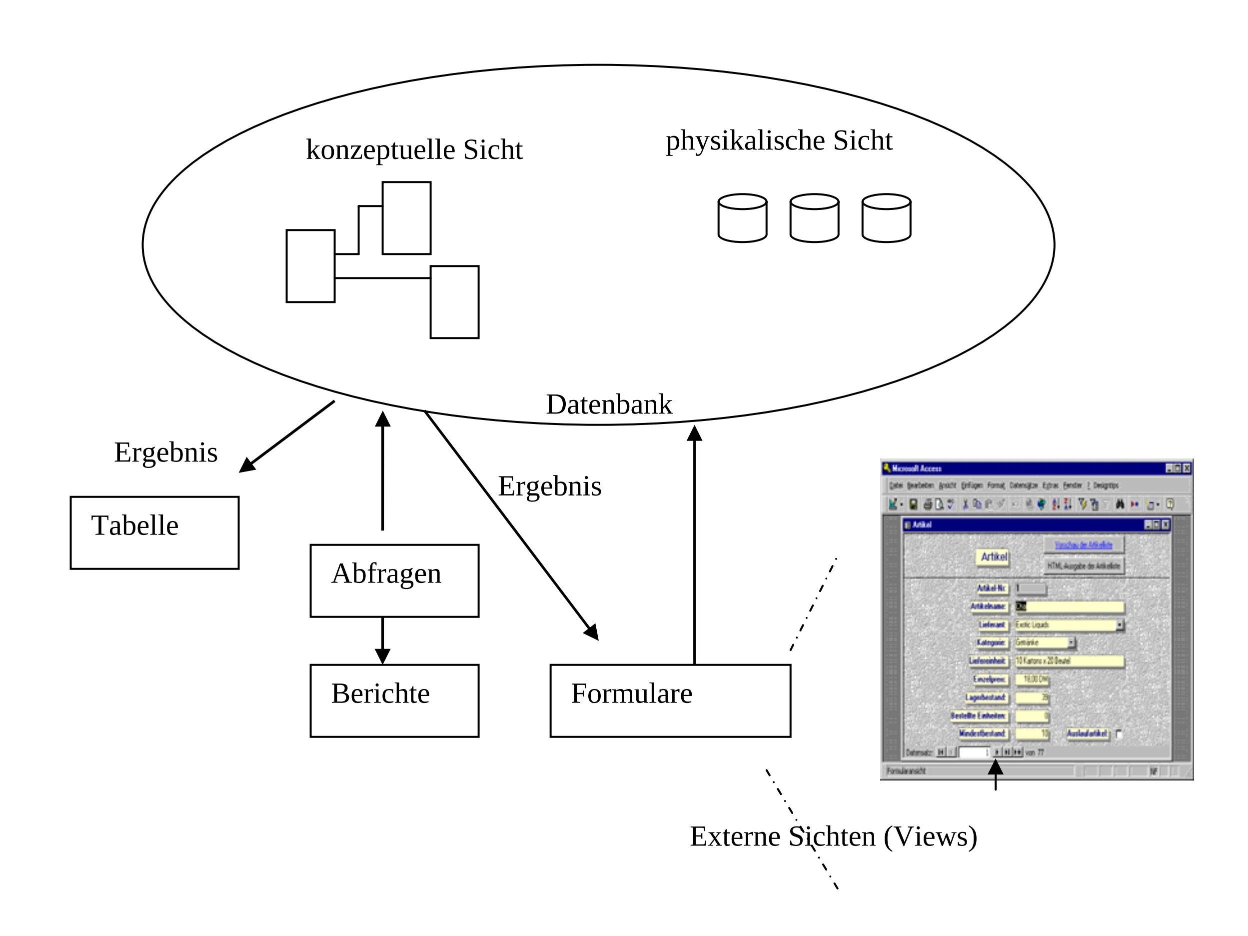
Hier sind die logischen Zusammenhänge und Abhängigkeiten der Daten beschrieben. Die Abstraktion, die von der realen Welt zu dieser logischen Gesamtsicht führt, bezeichnet man auch als konzeptuelles Modell.

Das relationale Datenmodell ist das momentan am weitesten verbreitete. Es basiert rein auf Tabellen (Relationen) und deren Verknüpfungen. Ein Ausschnitt der Welt (das Problem) wird Problem angemessen modelliert (ER-Modell) und dann auf das relationale Schema abgebildet. Dieser Prozess der Modellbildung, sprich Datensammlung, Aufteilen in Tabellen, Bestimmung der entsprechenden Datentypen und die Verknüpfung der Tabellen ist das Datenmodell, die konzeptuelle Sicht der Datenbank.

Interne Sicht:

Hier wird die Realisierung der Daten auf einer Computeranlage beschrieben. Es geht um Fragen, wie: Welche Daten werden zu Einheiten (Datensätzen) auf einem Massenspeicher zusammengefasst und wie wird schnell darauf zugegriffen (Suchbaum, Hashtabelle usw.)





Im Idealfall sind die drei Ebenen völlig unabhängig voneinander. Das bedeutet z.B., dass eine andere interne Organisation der Daten nichts am konzeptuellen Modell ändern sollte.

Die Verbindungen werden durch Transformationen geschaffen, die natürlich bei Änderungen auf einer Ebene anders gestaltet werden müssten.

Die Nutzer haben demnach auf der externen Ebene Zugang zur Datenbank, während es eine Instanz (und Person) gibt, die ein konzeptuelles Modell des Unternehmens erstellt und sich um die interne Organisation der Daten kümmert.

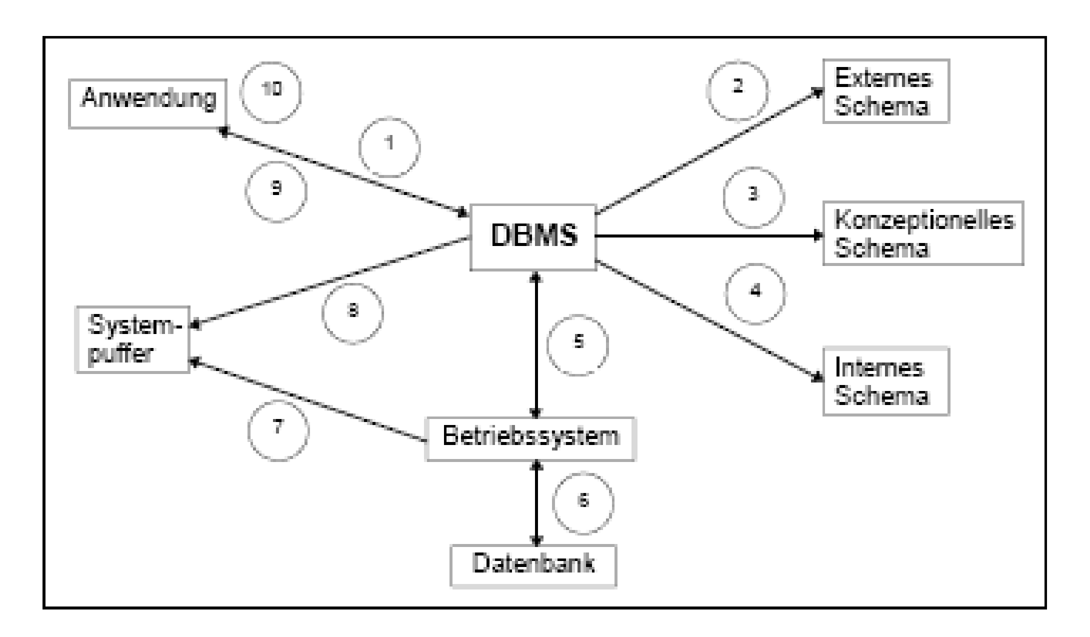
Auf das interne Modell soll nicht weiter eingegangen werden, obwohl natürlich entsprechende Fragen in der Informatik ihre Bedeutung haben. Die interne Sicht wird vom RDBMS weitgehend verborgen.

In einem Unternehmen können Systeme nur sinnvoll in einer vernetzten Umgebung ablaufen, d.h. das DBMS muss im Regelfall mehrere von außen angeforderte Operationen quasi gleichzeitig bearbeiten können. Dies bedeutet, dass es dem einzelnen Benutzer so erscheint, als würde er allein mit dem System arbeiten, obwohl ganz unterschiedliche Benutzergruppen auf das System zugreifen. Dabei müssen spezielle Sicherheitsmaßnahmen beachtet werden. So könnte z.B. der Versuch des gleichzeitigen Eintragens der Noten für ein- und denselben Schüler an zwei verschiedenen Stellen eine Verletzung der Datenintegrität hervorrufen. Ein eigenes Programm des DBMS, der so genannte *Transaktionsmanager*, soll das verhindern. Dabei sind die verschiedensten Architekturen denkbar. So könnte ein Zentralgerät (Server) nicht nur die Daten, sondern auch die wesentlichen Teile des DBMS enthalten, auf den angeschlossenen Arbeitsstationen (Clients) müsste dann lediglich ein Programm für Anfragen an die Datenbank ablaufen. Umgekehrt wäre es denkbar, dass auf jedem Client ein DBMS arbeitet und der Server nur als zentraler Datenspeicher dient. Prinzipiell wäre auch eine nahezu beliebige Verteilung von Daten und DBMS über die Clients ohne speziellen Server möglich (verteilte Systeme), wobei dem Benutzer allerdings das Ganze weiterhin als zentrale DB erscheint. Die Realisierung, sowie die Vor- und Nachteile dieser verschiedenen Architekturen, bilden wieder ein eigenes Kapitel der Informatik. Es sei jedoch nur angemerkt, dass *Access* in lokalen Netzen jeweils auf den Clients läuft und den Server nur als Datenspeicher benötigt, wobei allerdings eine gewisse Transaktionsüberwachung, das so genannte „Record-Locking“, stattfindet. Es gibt aber auch die Möglichkeit, dass *Access* über eine normierte Abfragesprache auf bestimmte zentralisierte DBMS (so genannte SQL-Server) zugreifen kann.

Aufgabe:

Welche Anforderungen werden – vor dem Hintergrund des just erworbenen Wissens – also an eine Datenbank und ihr Managementsystem (DBMS) gestellt? Denken Sie auch darüber hinaus. Was passiert bei mehreren Benutzern, wie werden Daten gesichert usw.

Datenbankverwaltungssysteme (DBMS) stellen also die Verbindung zwischen der Datenbasis und den Datenbankbenutzern bzw. Anwendungsprogrammen her. Dabei erfolgt der Zugriff der Anwendungen auf die Datenbasis nicht direkt, sondern nur über das DBMS. Das folgende Diagramm veranschaulicht, wie das Datenbankverwaltungssystem eine Anfrage (Query) abarbeitet.



1. Das DBMS empfängt den Befehl des Anwendungsprogrammes, ein bestimmtes Objekt zu lesen.
2. Das DBMS holt sich die benötigten Definitionen des entsprechenden Objekttyps aus dem zugehörigen externen Schema.
3. Das DBMS stellt fest, auf welche konzeptionellen Objekte sich die Anfrage bezieht.
4. Das DBMS stellt fest, welche physischen Objekte zu lesen sind.
5. Das DBMS übergibt dem Betriebssystem die Nummern der zu lesenden Speicherblöcke.
6. Das Betriebssystem greift auf die physischen Speicherblöcke in der Datenbank zu.
7. Das Betriebssystem übergibt die verlangten Blöcke an das DBMS in einen Systempuffer.
8. Das DBMS stellt aus den vorhandenen physischen Sätzen im Systempuffer das verlangte Objekt zusammen.
9. Das DBMS übergibt das Objekt dem Anwendungsprogramm in seinen Arbeitsspeicher.
10. Das Anwendungsprogramm verarbeitet die vom DBMS übergebenen Daten.

2.3 Exkurs: Andere Datenmodelle

Neben dem relationalen Modell kann auf konzeptioneller Ebene auch nach anderen Methoden vorgegangen werden. Es gibt hier beispielsweise das:

- hierarchische Modell:
Beziehungen werden durch eine Baumstruktur dargestellt. Ausgehend von der Wurzel werden die Daten hierarchisch in Teilbäume gegliedert und strukturiert. Dabei hat jedes Datenobjekt genau einen Vorgänger.

- Netzwerkmodell:
hier kann ein Datenobjekt auch mehrere Vorgänger haben

- objektorientiertes Modell:
ähnlicher Ansatz wie in der objektorientierten Programmierung

Da in der Praxis momentan nur das relationale Modell Anwendung findet werden wir im Folgenden auch nur selbiges betrachten.

Aufgaben

3-Ebenen-Architektur von Datenbanksystemen

Aufgabe 1:
Es soll eine Datenbank für eine Bücherei eingerichtet werden. Ein einzelnes Buch sei beschrieben durch die folgenden Merkmale:
Inventarnummer des Buches, ISBN-Nummer, Autor, Titel, Fachgebiet, Verlag, Erscheinungsort und -jahr, Auflage, Preis.

a) Zu welcher Ebene gehört diese Beschreibung?

Additum:
Nehmen wir an, die Datenbank soll in Pascal/Java/… geschrieben werden. Welche Hilfsmittel stehen dann zur Verfügung, um möglichst saubere Schnittstellen zwischen den einzelnen Ebenen zu realisieren? Welche Design-Entscheidungen bei der Planung des Programms gehören zur physischen, welche zur konzeptuellen, welche zur externen Ebene? Inwieweit lässt sich Datenunabhängigkeit wahren?

Aufgabe 2:
Analysieren Sie bei jeder der folgenden Umstellungen, auf welcher der Ebenen einer bestehenden Datenbank und an welchen Transformationen jeweils etwas geändert werden muss (im Idealfall):

a) Neuer Rechner, aufwärts kompatibel, gleiches Betriebssystem, gleiches Datenbanksystem
b) Neuer Rechner, anderes Fabrikat, gleiches Betriebssystem, gleiches Datenbanksystem.
c) Neue Platten mit anderer Struktur
d) Ein neues Programm benutzt bestehende Daten.
e) Ein neues Programm X benutzt bestehende Daten und zusätzliche neue Datenstrukturen.
f) Eine neue Verbindung zwischen zwei existierenden Typen wird eingeführt, bisher gab es z.B. zwischen Schülern und Lehrern nur die Verbindung, wer Tutor ist. Jetzt gebe es zusätzlich die Verbindung, wer für jeden Schüler frei wählbarer Vertrauenslehrer ist.
g) Der Bestand einer Datenbank wird geteilt.

2.4 Von der Realität zum Modell

Informationssysteme, z.B. zur Kontoführung bei Banken, Auskunftsystemen für Flugzeuge und Bahnen, Börsenauskunftsysteme, Lagerhaltungssysteme usw., speichern und verwalten Informationen über einen bestimmten Ausschnitt der realen Welt. Aber selbst solche Ausschnitte sind meist so komplex und vielgestaltig, dass sie nicht alle Einzelheiten beschreiben bzw. dass über den vielen Einzelheiten der Überblick über das Wesentliche verloren gehen kann Deshalb ist oft bei der Beschreibung eine Beschränkung auf die wichtigsten oder interessantesten Objekte der Realität und ihre Beziehungen untereinander sinnvoll. Wir erzeugen dadurch einen Ausschnitt der realen Welt, eine *Miniwelt.* Beispielsweise ist die Schule ein Teil der realen Welt. Unter dem Blickwinkel der Verwaltung von Schülerdaten würde unsere Miniwelt aus den Objekten *Schüler*, *Lehrer*, *Bücher*, *Kurse, Noten* und deren Beziehungen zueinander bestehen; andere Objekte der Schule, wie Gebäudedaten, Reinigungskosten usw. würden ausgeklammert. Wir bilden deshalb im Hinblick auf die logische Gesamtsicht ein *Modell der Miniwelt,* Innerhalb des Modells können wir Fragen stellen, die im Modell beantwortet werden. Die Antworten sind aber durch das Modell begrenzt: So können wir keine Antwort darauf erhalten, wieso ein Schüler im Abitur gescheitert ist **-** hierzu haben wir weder Daten noch Methoden im Modell zur Verfügung. Wir müssen uns also immer bewusst sein, in welchem Bereich gültige Antworten zu erwarten sind und welche Bedeutung sie in der Wirklichkeit haben.

Innerhalb des Modells lassen sich Informationen sammeln, speichern und später wiedergewinnen bzw. durch Verknüpfung dieser Informationen neue Informationen finden, z.B. die Berechnung des Abiturdurchschnitts aus den Einzelnoten. Da unsere Informationen in einem Computersystem gespeichert werden, also zur Verarbeitung dienen, bezeichnet man sie auch als *Daten*. Die Daten sind Abbildungen des Modells der Miniwelt in den Speicher eines informationsverarbeitenden Systems.

Damit unser Modell ein möglichst getreues Abbild unserer Miniwelt darstellt, ist es vor allem notwendig, die realen Objekte und ihre Beziehungen zueinander möglichst exakt zu erfassen und zu beschreiben. Erst dann ist sichergestellt, dass das Modell nur solche Aussagen liefert, die wieder ihre Entsprechung in der Wirklichkeit haben. Wenn das Modell fehlerhaft ist, z.B. wegen fehlender oder falscher Daten oder falscher Struktur der Daten und ihrer Beziehungen, dann können nicht immer zuverlässige Aussagen über die reale Welt gemacht werden. Eine unserer wichtigsten Aufgaben wird also darin bestehen, Methoden zur Modellierung von Realität zu entwickeln, damit unser Modell zuverlässige Aussagen liefert.

Die Informationen über eine solche Miniwelt, z.B. die Schulverwaltung, erhalten wir am besten durch Sichtung der Unterlagen, mit der sie sich nach außen dokumentiert: Durch Formulare wie Datenblätter, Zeugnisse, Belegunterlagen usw. beschreibt sich die reale Welt. Wir beschränken uns bei ihrer Beschreibung auf die Tatsachen, die unter dem Zweckbezug und Aspekt der Schulverwaltung notwendig und wesentlich sind und damit unsere Miniwelt repräsentieren. Alle anderen Tatsachen sollen unberücksichtigt bleiben mit der Konsequenz, dass unser Modell hierfür keine zuverlässigen Aussagen liefern kann. Das dargestellte Zeugnisformular einer Schule soll als Ausgangspunkt dienen.

Bei eingehender Prüfung können wir feststellen, dass Zeugnisse Informationen über verschiedene *Objekte* der realen Welt bzw. unserer Miniwelt liefern, wie z.B. *Schüler*, *Kurse*, *Schule*. Andere Informationen beziehen sich nicht auf ein Objekt allein, sondern stellen Eigenschaften einer *Beziehung zwischen Objekten* dar: *Schülerin* besucht *Kurs*. Damit können wir den speziellen Ausschnitt der realen Welt, unsere Miniwelt Schulverwaltung, im Modell beschreiben:

Internatsgymnasium Schloss Neuschwanstein
Wiesbaden
Jahrgangsstufe 12 (Qualifikationsphase)   Schuljahr 2009/2010, 1. Halbjahr

Z E U G N I S
Tina Musterfrau

Fach	Kurs	WStd	Kursthema	Kursleiter	Punkte
Sprachlich-literarisch-künstlerisches Aufgabenfeld					
Deutsch	Gk	3	Ideal und Wirklichkeit	Klaas	11
Englisch	Lk	5	Begnung der 3. Art	Meyer	04
Musik	Gk	3	Flötenspiele	Meyer	06
Gesellschaftswissenschaftliches Aufgabenfeld					
Gemeinschaftskunde	Gk	3	Sozialordnung	Schulz	07
Geschichte	Gk	3	III. Reich	Herberg	05
Mathematisch-naturwissenschaftlich-technisches Aufgabenfeld					
Mathematik	Lk	5	Analysis	Wegener	06
Biologie	Lk	5	Molekularbiologie	Klever	03
Chemie	Gk	3	Organische Chemie	Weber	03
Sport	Gk	3	Volleyball	Abt	00

Fehlstunden: 22   davon 4 unentschuldigt

Bemerkungen: keine

___Merten_______	___Wegener______	____Tina ____
Turor(in)	Lehrer(in)	Schüler(in)

Datum: 30.01.2010

Informationen zu unserem Modell

Allgemeine Struktur	Konkrete Ausprägung
Schule Schulname Schulort	 Schloss Neuschwanstein Wiesbaden
Schüler(in) Name Vorname Tutor(in)	 Musterfrau Tina Merten
Zeugnis Halbjahr Datum	 1. Halbjahr 2009/2010 30.01.2010
Kurs Fach Kursthema Kurstyp Lehrer(in)	 Mathematik Analysis LK Wegener
Schüler(in)-Kurs Punkte	 06
Schüler(in)-Zeugnis Fehlstunden entschuldigt Fehlstunden unentschuldigt Bemerkungen	 22 4 keine

Informationen wie *Schulname*, *Tutor Halbjahr*, stellen **Eigenschaften** der betreffenden **Objekte** Schule, Schüler bzw. Kurs dar. Die Informationen zur Note oder über Bemerkungen beziehen sich dagegen auf jeweils zwei Objekte gemeinsam, d.h. sie stellen Eigenschaften einer **Beziehung** zwischen den Objekten Schüler(in)/Kurs bzw. Schüler(in)/Zeugnis dar. Wir können diesen Beziehungen die Namen „besucht“ und „erhält“ geben, es entstehen also die Beziehungen

- Schüler(in) *besucht* Kurs
- Schüler(in) *erhält* Zeugnis

zwischen jeweils 2 Objekten.

In der Informatik spricht man im Zusammenhang mit Datenbanksystemen meist nicht von Objekten, sondern benutzt den Begriff *Entity* oder auch *Entität*. Entities können demnach Personen sein, reale Objekte wie Zeugnisse oder Räume, aber auch abstrakte Objekte, wie z.B. Kurse, die nur gedanklich als ein unterscheidbares und identifizierbares Objekt existieren. Eine Entity wird im Wesentlichen durch seine Eigenschaften beschrieben, im konkreten Fall besitzt jede Eigenschaft einen Wert, z.B. die Eigenschaft „Kursbezeichnung“ der Entity „Kurs“ den Wert *3M11*. Dies entspricht dem Klassen-Objekt Prinzip in der objektorientierten Programmierung. Die Attribute (Spalten) einer Tabelle, ihre Eigenschaften, bilden die Klasse, während die konkreten Daten das Objekt bilden. Wir werden noch genauer untersuchen, wie man Entities und ihre Beziehungen zueinander beschreiben und darstellen kann. Die folgende Darstellung zeigt aber bereits alle wesentlichen Merkmale unserer Miniwelt, wie sie sich durch die Zeugnisformulare dokumentiert:

Entity	Beziehung	Eigenschaften
Schule		Schulname Schulort
Kurs		Kursbezeichnung Typ Lehrer
	Schüler(in) *besucht* Kurs	Note
Schüler(in)		Name Vorname Tutor
	Schüler(in) *erhält* Zeugnis	Fehlstunden entschuldigt Fehlstunden unentschuldigt Bemerkung
Zeugnis		Halbjahr Datum

Die bisher ausgewerteten Informationen über die Schulverwaltung reichen bei weitem nicht aus, ein getreues Abbild unserer Miniwelt zu erzeugen. In Abhängigkeit vom Zweck der Modellbildung müssen weitere Dokumente ausgewertet werden. Für eine automatisierte Kurswahl müssen beispielsweise u.a. Kursthemen, Räume und Zeiten erfasst werden. Sollen Zeugnisse einschließlich des Abiturzeugnisses gedruckt werden, sind sämtliche Kursbelegungen während der Oberstufe mit Noten und Fehlstunden aufzunehmen. Für eine Bibliotheksverwaltung sind Informationen über den Buchbestand und das Ausleihverfahren einzuholen. Die jeweiligen Datenmodelle können dann als Abbild einer Miniwelt dienen, die einen Teilbereich der Schulverwaltung abdeckt.

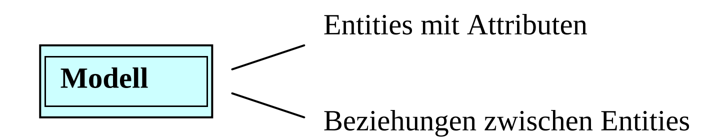
Um die jeweilige Miniwelt genauer zu beschreiben, ist über die Angabe der Objekte (Entities) und ihrer Beziehungen zueinander hilfreich, so genannte *Geschäftsregeln* festzuhalten. Sie stellen nichts anderes als Feststellungen über die Objekte der Miniwelt dar:

Ein Zeugnis geht genau an ein(en) Schüler(in) und existiert ohne diesen nicht. Ein(e) Schüler(in) kann im Laufe der Oberstufe mehrere Zeugnisse erhalten; es gibt auch Schüler(innen), die noch kein Zeugnis erhalten haben, z.B. in der Jahrgangsstufe 11-I oder als Neuzugang.
Ein Zeugnis enthält mindestens eine Kursangabe mit Note und Angaben über Fehlstunden und eventuell Bemerkungen; ohne Zeugnis sind diese Angaben sinnlos.
Eine Kursangabe bzw. Fehlstunden und Bemerkungen können in vielen Zeugnissen auftauchen; es gibt Bemerkungen, die auf keinem Zeugnis stehen.

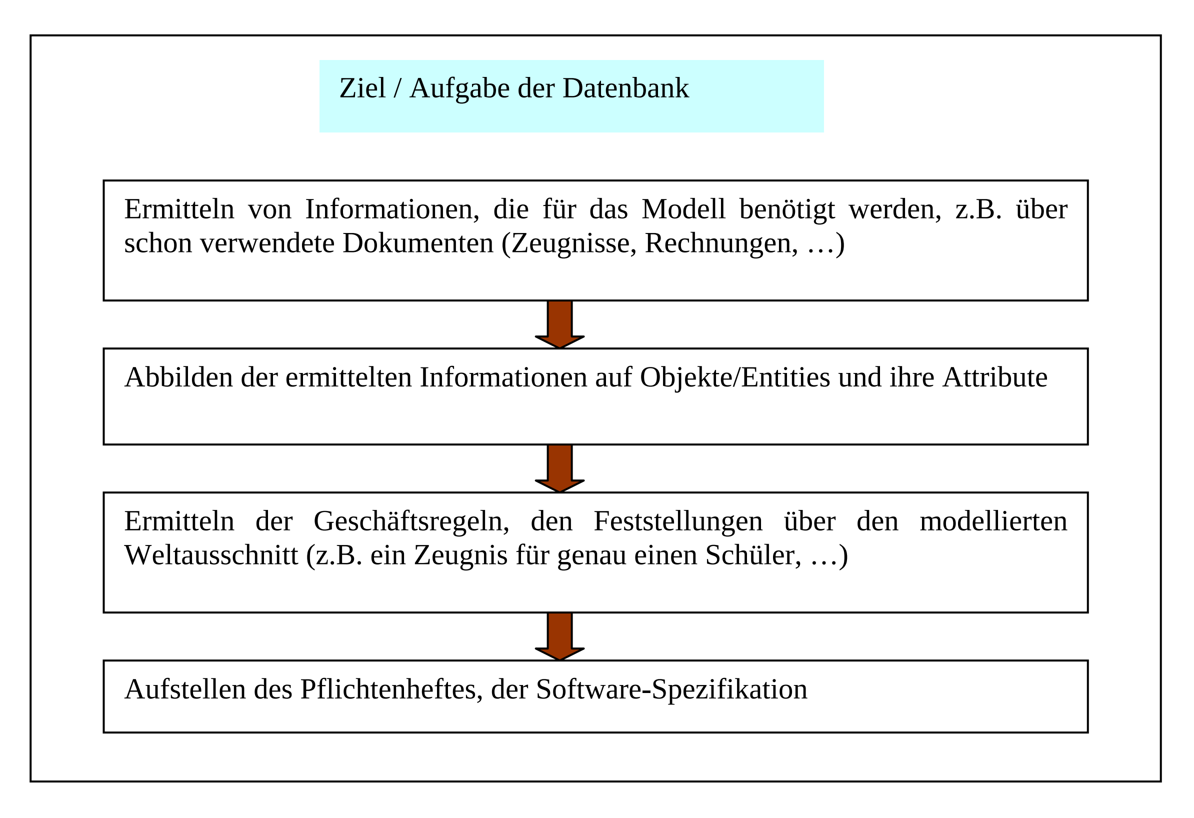
Wir werden später sehen, dass das Modell unserer Miniwelt umso besser der Realität entspricht, je genauer die Geschäftsregeln festgelegt sind. Beim Aufbau eines Datenbanksystems wird dadurch u.a. die Datenintegrität sichergestellt. Es ist dabei von Bedeutung, ob ein Objekt, z.B. ein Zeugnis, ohne ein anderes Objekt, z.B. eine Schülerin, existieren kann oder nicht.
Die Geschäftsregeln entsprechen einem Teil des *Pflichtenheftes* im Softwareengineering. Dort wird genau festgehalten, was das Softwareprodukt können soll. Dazu ist eine genaue Bestandsaufnahme des Realweltausschnitts notwendig, ebenso wie eine Einteilung des Gesamtproblems in Teilprobleme und in eine zeitliche Reihenfolge, die durch Meilensteine, also Teilziele, fixiert wird.

Fassen wir zusammen:

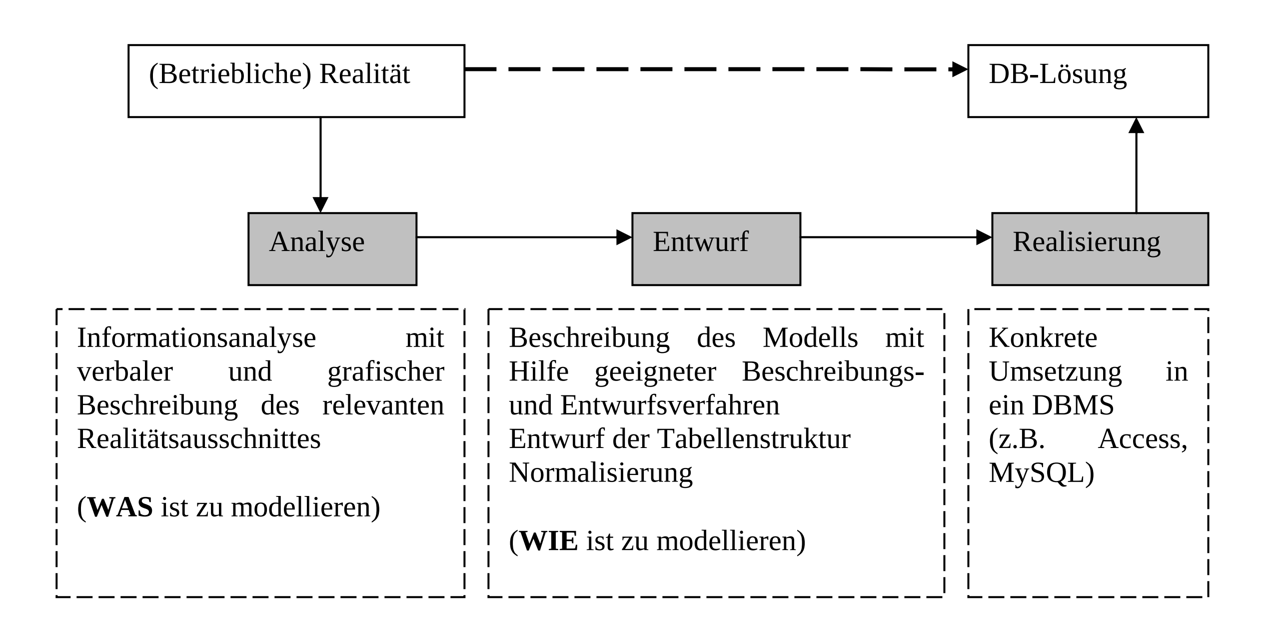
Ein Modell besteht aus Objekten/Entitäten mit ihren Eigenschaften (Attributen), die in Beziehung zu einander stehen.



Der Weg zum Modell, ausgehend von der Aufgabenstellung/dem Ziel, erfolgt schrittweise.



Also



Aufgabe 1:
Was verstehen Sie generell unter dem Begriff der ’Modellbildung’ in der Informatik/Technik?

Aufgabe 2:
Erläutern Sie die Begriffe Objekt, Entity, Eigenschaften, Attribute und Beziehungen.

Aufgabe 3:
Wovon geht man zu Beginn der Modellbildung i.d.R. aus?

Aufgabe 4:
Nehmen Sie sich eine Mini-Welt Ihrer Wahl vor und überlegen Sie sich geeignete Inhalte für die Struktur.

Entity	**Beziehung**	**Eigenschaften**
…	…	…
…	…	…

Aufgabe 5:
Sie bekommen von einem Stahlwerk den Auftrag, den Schmelzvorgang für besondere Stahlsorten datenbanktechnisch zu erfassen. Beschreiben Sie generell und an Hand der kennen gelernten Fachbegriffe und Vorgehensmodelle, wie Sie an das Problem herangehen.
.

2.5 Miniwelt “Informatik-Bibliothek“ – Anforderungsprofil und Geschäftsregeln

Die Modellschule Obersberg besitzt eine relativ große Bibliothek mit eigenen Angestellten, die für die Ausleihe sowohl der Einzelbücher als auch der Klassensätze zuständig sind. In den letzten Jahren hat sich im Bereich Informatik ein größerer Buchbestand angesammelt. Die Bibliothek arbeitet bisher ohne EDV. In die Schule integriert ist eine kaufmännische Berufsschule, die traditionsgemäß ihre Bücher selbst verwaltet. Auch dort existiert einiges im Bereich Informatik, vor allem zu Anwendungsprogrammen und Wirtschaftsinformatik. Die Informatiker der gymnasialen Abteilung haben sich nach und nach einen Handapparat an Büchern zugelegt, der auf Wunsch an interessierte Lehrer und Schüler ausgeliehen wird. Im Zuge einer Neuausstattung mit Hard- und Software wurde darauf geachtet, möglichst viele Programme mit Handbuchsätzen zu erwerben, um auf diese Weise den Schülern zusätzliche Hilfen an die Hand geben zu können. Damit hat die von den Informatikern verwaltete Zahl von Büchern die Tausendergrenze nahezu erreicht.

Die bisher praktizierte Ausleihmethode, dass jeder Lehrer, der sich oder einem Schüler ein Buch auslieh, einfach ein kleines Zettelchen in den Schrank legte, führt bei dieser Größe zu Chaos. Bücher bleiben verschwunden, ohne dass feststellbar ist, wohin. Kollegen betrachten die mitgenommenen Bücher als Dauerleihgaben, so dass bestimmte Titel, wenn überhaupt, erst nach Jahren wieder auftauchen. Schüler gehen von der Schule ab, ohne dass die Bücher zurückgefordert worden wären usw.

Deshalb entsteht die Idee, die Ausleihe per EDV zu organisieren. Gleichzeitig sollen in einem solchen System aber auch alle weiteren Informatikbücher der Schule erfasst werden, für deren Ausleihe nicht die Informatikabteilung zuständig ist, um so allen Interessierten eine schnelle Übersicht über die vorhandenen Titel des Fachgebietes zu ermöglichen. Mit der Realisierung dieser Informations- und Ausleihdatenbank beschäftigt sich das Ihnen vorliegende Projekt.

Kritiker werden hier zu Recht anmerken, dass es, anstelle eines speziellen Ausleihprogramms der Informatiker, sinnvoller wäre, die interne Organisation der Schulbibliotheken zu ändern und alle Bücher an einem zentralen Ort zu sammeln. Wir wollen hier jedoch von den *momentanen Gegebenheiten* ausgehen, die bekanntlich auch andernorts oft ein *zähes Eigenleben* besitzen. **Nicht der Einsatz des Computers allein kann immer die Organisation entscheidend verbessern**.

Die Spezifikationen der zu erstellenden Softwarelösung könnte als “Pflichtenheft“ beispielsweise wie folgt aussehen:

> Es sollen alle Informatikbücher der Modellschule Obersberg in einer Datenbank erfasst werden.
>
> Informatikbücher gibt es im Moment in der Mediothek zur Individualausleihe und in der Lernmittelbücherei. Weiterhin gibt es Informatikbücher in den Informatikräumen der gymnasialen Abteilung und zwar als Einzelexemplare und in jeweils größerer Stückzahl. Auch die kaufmännische Abteilung verwaltet in Eigenregie Bücher aus diesem Themengebiet. Weitere mögliche Standorte sollen vorgesehen werden.
>
> Es sollen über ein Buch gespeichert werden: Autor, Titel, Themengebiet, Ort der Aufbewahrung, Ausleihstatus (Präsenz, übers Wochenende, vier Wochen, halbes Jahr), Erscheinungsjahr, Verlag, Stichworte (bis zu 20 pro Buch), Kurzbeschreibung. Eine Suche soll nach Themengebiet, Autor, Titel und Stichwort möglich sein.
>
> Weiterhin soll mit dieser Datenbank die Ausleihe in den Informatikräumen der gymnasialen Abteilung organisiert werden. Dazu sollen die Ausleiher erfasst werden (Schüler und Lehrer) mit Namen und Adresse, bei den Schülern außerdem Klasse und Tutor. Auf Lesekarten und Ausweise kann verzichtet werden. Die Ausleihe soll von den Lehrern durchgeführt werden. Lehrer und Schüler können mehrere Bücher ausleihen. Die meisten Bücher sind schon mit einer Inventarnummer versehen (10-stellige Zahl). Es muss möglich

sein, neue Entleiher zu erfassen, bzw. alte zu löschen. Entleih- und Rückgabevorgänge müssen durchgeführt werden können. Das Programm soll ausgeben können: Entleiher eines bestimmten Buchtyps, alle entliehenen Bücher eines bestimmten Entleihers, Mahnungen an alle Entleiher, die ihre Frist überschritten haben.

Das so zusammengestellte Anforderungsprofil könnte nach gründlicher Durchsicht z.B. noch folgende Änderungen annehmen:

(1) Auf die Speicherung der Adresse kann verzichtet werden, da die Schüler leichter über ihren Tutor zu erreichen sind (zudem keine Postgebühren) und die Lehrer Fächer im Lehrerzimmer besitzen. In Ausnahmefällen kann auf die Adresse im Sekretariat zurückgegriffen werden.

(2) Beim Entleihen sollte ein Zettel für den Entleiher gedruckt werden, der neben dem Titel des Buches auch ein Rückgabedatum enthält.

(3) Die Datenbank sollte auf dem Server des Netzwerkes geführt werden und von jedem Rechner mit jeder Zugangsberechtigung eine Suche nach Buchtiteln möglich sein.

(4) Die Ausleihe kann nur mit einem Lehrerpasswort durchgeführt werden. Die Frage, ob das auf jedem Rechner möglich sein soll oder nur auf einem speziellen dafür vorgesehenen, wird zunächst Offengelassen.

(5) Für die Bücher an „fremden“ Standorten ist es nicht notwendig, Signaturen zu speichern, da ohnehin keine Ausleihe von der zu entwickelnden Software durchgeführt werden soll. Es würde bei diesen Büchern ausreichen, neben den allgemeinen Angaben und dem Standort die Anzahl der maximal vorhandenen einzugeben, damit der Sucher eine Vorstellung hat, wo und wie oft das Buch vorhanden ist.

Mit diesen Festlegungen können die Entities und die Beziehungen zwischen ihnen bereits angegeben werden:

Objekt	Beziehung	Eigenschaften
Ausleiher		Name Vorname Klasse (bei Schülern) Tutor (bei Schülern)
	Ausleiher *leiht aus* Buch	Ausleihdatum
Buch		Inventarnummer Autor(en) Titel Erscheinungsjahr Verlag Thema Kurzbeschreibung Stichwort(e) Standort Ausleihzeit

Diese Beschreibung enthält noch einige Problem: Die Entität *Ausleiher* ist z.B. nicht eindeutig, da Schüler und Lehrer unterschiedliche Eigenschaften besitzen. Hierzu werden wir im nächsten Abschnitt das Entity-Relationship-Modell genauer untersuchen und auf unser Beispiel anwenden.

Die Geschäftsregeln unserer Miniwelt lassen sich aus der Ist-Analyse und durch Analogiebildung zum Schulverwaltungsbeispiel leicht aufstellen:

- Ein Buch kann keinmal, einmal oder mehrfach ausgeliehen werden. Es gibt auch Bücher, die nicht ausgeliehen werden dürfen.
- Ein Ausleiher kann kein Buch oder ein Buch oder mehrere Bücher ausleihen. Ausleiher und Bücher existieren unabhängig voneinander.
- Eine Ausleihe enthält mindestens Angaben über den Ausleiher, das ausgeliehene Buch und das Ausleihdatum.

2.6 Das Entity-Relationship-Modell

Wie im vorletzten Kapitel ausgeführt, ist es zunächst sinnvoll, ein **konzeptuelles Modell** des gegebenen Problems zu erstellen. Dabei wird **ganz allgemein ein Abbild der Wirklichkeit geschaffen**, bei dem bestimmte Aspekte weggelassen, vereinfacht, zusammengefasst werden. Es lassen sich dabei drei **Abstraktionsmechanismen** unterscheiden:

Klassifikation: Dinge bzw. Objekte mit gemeinsamen Eigenschaften werden zu einer *Klasse* zusammengefasst. Dabei werden nicht alle Eigenschaften der Objekte, meist *Attribute* genannt, berücksichtigt, sondern nur die für die Aufgabe relevanten. In unserer Miniwelt Schulverwaltung lässt sich das Objekt „Schüler" als Klasse auffassen, in der Miniwelt Bibliothek das Objekt „Buch". Die zugehörigen Eigenschaften oder Attribute haben wir bereits definiert.

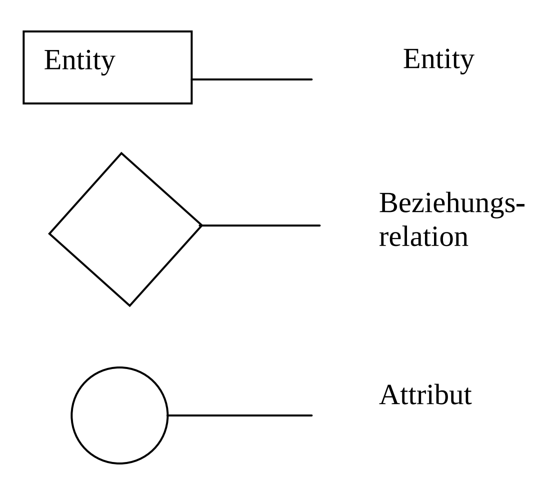
Aggregation: Bereits bestehende Klassen werden zu einer neuen Oberklasse zusammengefasst. So würde sich „Abiturbewertung" aus „Halbjahresnoten" und „Prüfungsergebnisse" zusammensetzen.

Generalisierung oder *Spezialisierung*: Dabei wird eine Teilmengenbeziehung zwischen Elementen verschiedener Klassen definiert. So wäre „Ausleiher in der Schulbibliothek" eine Generalisierung der Klassen „Schüler" und „Lehrer". Umgekehrt wäre „Abiturient" eine Spezialisierung von „Schüler". Bei Generalisierung, bzw. Spezialisierung gibt es immer *Vererbung*. So besitzt die untergeordnete Klasse (die Teilmenge) alle Attribute der Oberklasse, kann aber noch zusätzliche Eigenschaften haben. „Ausleiher" braucht sicher die Attribute „Name, Vorname", die Teilmenge Schüler besitzt noch „Noten", „Lehrer" besitzt zusätzlich noch „Unterrichtsfächer".

Um die Modellierung der Realität systematischer gestalten zu können, gibt es verschiedene Standardstrategien. Eine der bekanntesten zur Erstellung von Datenbank-Anwendungen ist das so genannte Entity-Relationship-Modell (kurz ER-Modell genannt). Das ER-Modell wurde zuerst von P.P. Chen 1976 vorgeschlagen und verknüpft Objekte (Entities) und ihre Beziehungen (Relationships) zueinander in einem gemeinsamen Modell. Dabei versteht man unter einem Entity ein konkretes Objekt der Realität, z.B. den Schüler Fritz Immerfaul, unter einem Entity-Typ eine sinnvolle Klasse von einzelnen Entities, in unserem Beispiel wäre das der Typ „Schüler".

Anm.: In der objektorientierten Programmierung (OOP) ist das modellierte Objekt durch eine *Klasse* formuliert, sozusagen die Blaupause, die Vorlage, aus der alle konkreten Instanzen entstehen. Selbige konkrete Ausprägungen der Klasse werden in der OOP *Objekte* genannt, ihre Attribute *Eigenschaften*.

Zur Modellierung werden im Entity-Relationship-Modell drei Einheiten verwendet:



Die Entities eines Typs besitzen gemeinsame Eigenschaften, *Attribute* genannt, im Beispiel *Name*, *Vorname*, usw. Die Attribute können atomar sein, d.h. nur einen Wert umfassen, mehrere Werte ermöglichen oder selbst wieder aus einzelnen Attributen zusammengesetzt sein. Man kann ein Objekt mit seinen Attributen graphisch sehr gut veranschaulichen:

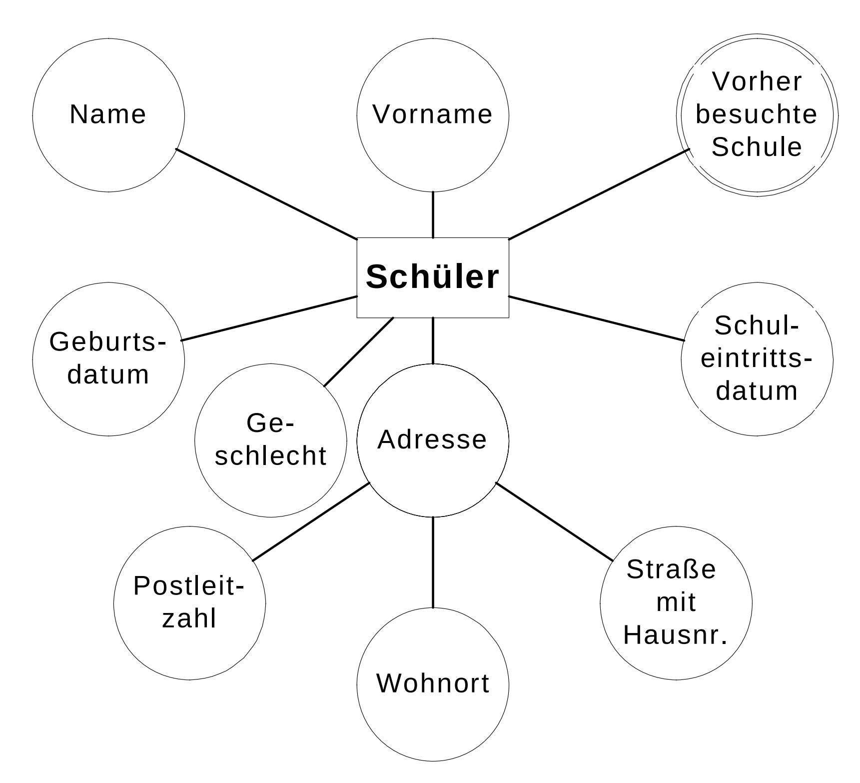


Abbildung: ER-Modell Schüler

Dabei wird der Entity-Typ durch ein Rechteck dargestellt, die Attribute durch mit dem Rechteck verbundene Kreise. Ein zusammengesetztes Attribut wie „Adresse“ wird durch entsprechende Folgekreise gezeichnet. Ein Attribut wie „Vorher besuchte Schule“, das einen oder mehrere Einträge haben kann, erscheint als Doppelkreis. Solche Attribute heißen *Mehrfachattribute.*

Anm.: Im relationalen Datenmodell werden Attribute dann durch Spalten(überschriften) dargestellt.

Ein oder mehrere Attribute, die eine Entity eindeutig charakterisieren, werden *Schlüssel* genannt. In unserem Beispiel wäre ein möglicher Schlüssel die Kombination der Attribute „Name, Vorname, Geburtsdatum“, denn es erscheint doch extrem unwahrscheinlich, dass es

zwei Schüler mit gleichem Nachnamen, Vornamen und Geburtsdatum gibt. Es ist denkbar, dass noch andere Attributmengen mögliche Schlüssel sind. Das Attribut, welche den Datensatz eindeutig identifiziert, wird *Primärschlüssel* (PS) genannt und *durch Unterstreichen der Attributnamen kenntlich gemacht*. Häufig verwendet man auch *künstliche Attribute*, um einen Primärschlüssel zu schaffen. In unserem Beispiel wäre das eine Schüler-Nummer, die jedes Entity eindeutig charakterisiert. Besteht der Primärschlüssel aus mehreren Attributen, so spricht man von einem *zusammengesetzten Primärschlüssel*.

Zwischen den einzelnen Entities können *Beziehungen (Relationships)* bestehen, zwischen Entity-Typen existieren entsprechend *Beziehungs-Typen.* Beziehungs-Typen werden grafisch durch Rauten dargestellt, in die man die Beziehung notiert. Auch Beziehungs-Typen können Attribute besitzen. Als Beispiel betrachten wir den Beziehungs-Typ „besucht“ zwischen den Entity-Typen „Schüler“ und „Kurs“, die „Note“ als Attribut hat.

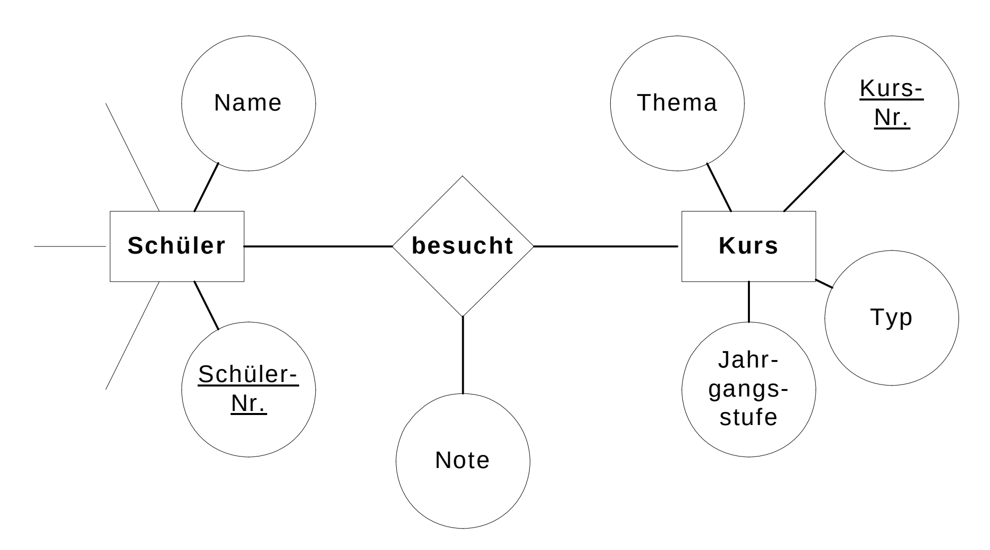


Abbildung: Beziehung im ER-Modell

Die Note gehört nicht zum Schüler und auch nicht zum Kurs, sie ist vielmehr ein Produkt der Beziehung zwischen dem Schüler und dem Kurs, den er besucht. Gäbe es die Beziehung nicht, entstünde auch keine Note, daher die Zuordnung zur Beziehungsrelation und nicht zu einer der beiden Entities.

Zwischen Entities kann es mehr als eine Beziehung geben, wie das folgende Beispiel zeigt.

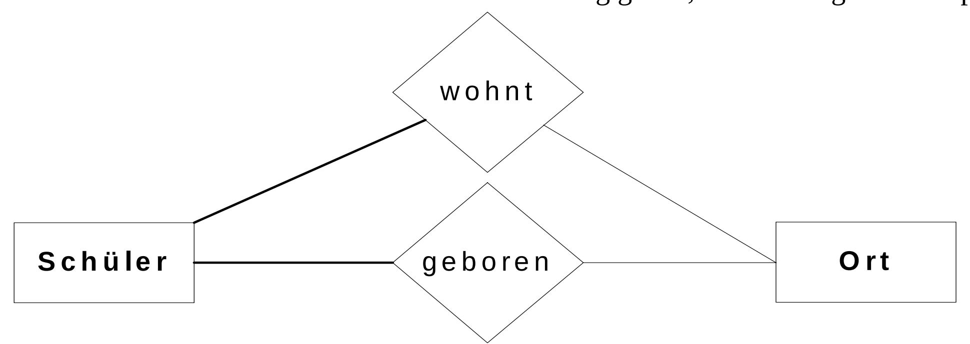


Abbildung: Mehrfachbeziehung

Ein Beziehungs-Typ kann sich auch auf mehr als zwei Entity-Typen beziehen: Verschiedene Großhändler liefern verschiedene Artikel an unterschiedliche Kunden. Die Lieferbeziehung ist also durch ein Tripel (Großhändler, Artikel, Kunde) gekennzeichnet.

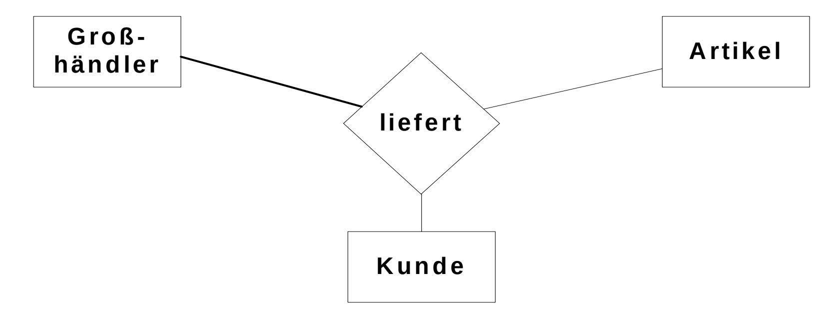


Abbildung 1: Dreiwertige Beziehung

Allerdings sind die zweiwertigen Beziehungen die weitaus häufigsten.

Die bisherige Darstellung sagt aber noch nichts über die Anzahl der jeweils an einer Beziehung beteiligten Entities aus. Im Normalfall unterrichtet z.B. ein Lehrer mehrere Kurse, ein Kurs besitzt aber nur einen Lehrer. Diese Tatsache beschreibt man mittels des Begriffs der *Komplexität* einer Beziehung: Die Beziehung *Lehrer unterrichtet Kurs* besitzt die Komplexität 1:n, da "1" Lehrer mehrere "n" Kurse unterrichtet.

Insgesamt unterscheidet man meist zwischen drei *Komplexitäten*:

1:1-Beziehungen: Jedes Entity vom Typ E1 steht höchstens mit einem Entity vom Typ E2 in Beziehung und umgekehrt (also mit einem oder keinem).

1:n-Beziehungen: Jedes Entity vom Typ E2 steht höchstens mit einem Entity vom Typ E1 in Beziehung, es können aber mehrere (oder keiner) aus E2 zum selben Entity aus E1 eine Beziehung haben. n:1-Beziehungen sind natürlich ganz analog definiert.

n:m-Beziehungen: Jede Entity aus E1 kann zu mehreren aus E2 eine Beziehung haben, und jede Entity aus E2 zu mehreren aus E1.

Beispiele:

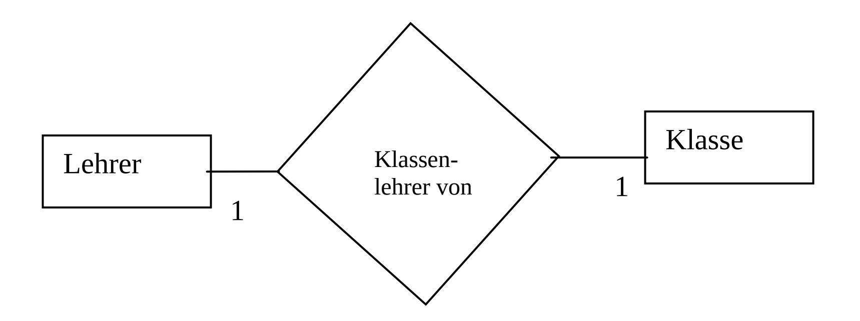
Komplexität 1:1: Schüler - erhält - Abiturzeugnis.
Jeder Schüler erhält (höchstens) ein Abiturzeugnis, und jedes Abiturzeugnis gehört eindeutig zu einem Schüler.

Komplexität 1:n: Lehrer - ist Tutor von - Schüler.
Jeder Schüler hat nur einen Tutor, aber jeder Lehrer, der Tutor ist, hat mehrere Schüler in seiner Tutorengruppe.

Komplexität n:m: Schüler - besucht - Kurs.
Jeder Schüler besucht mehrere Kurse, und jeder Kurs wird von mehreren Schülern besucht.

Manchmal findet man noch die Notation "c" für [0,1], also **1:c-Beziehungen**. Sie werden technisch wie 1:n Beziehungen behandelt.

Im ER-Diagramm werden die Komplexitäten **am gegenüber liegenden** Entitätstypen notiert. [**VORSICHT, FEHLERQUELLE!**]



lies: Jeder Lehrer unterrichtet 1 Klasse. Jede Klasse wird von einem Lehrer unterrichtet.

BEACHTE:
Gemäß der mathematischen Notation für Intervalle gibt es zwei mögliche Arten für die Komplexität 1:1. Es sind dies

[0,1] → höchstens einer und mindestens Null (also keiner), erlaubt
[1,1] → höchstens einer und mindestens einer, also genau einer

Eine besondere Art von Beziehungen sind die so genannten "**IS-A**" Beziehungen (aus dem Englischen, wörtlich übersetzt "ist ein"). Damit wird es möglich, Teilmengen von Entity-Typen zu erfassen. Diese Beziehungen entsprechen der oben angegebenen speziellen Abstraktionsform *Spezialisieren* bzw. *Generalisieren*. Bei bestimmten Entities gibt es sinnvolle Teilmengen, die alle Attribute der übergeordneten Menge erben (diese müssen nicht mehr in der grafischen Darstellung angegeben werden), aber noch zusätzliche Attribute besitzen. Ein Beispiel soll das verdeutlichen.

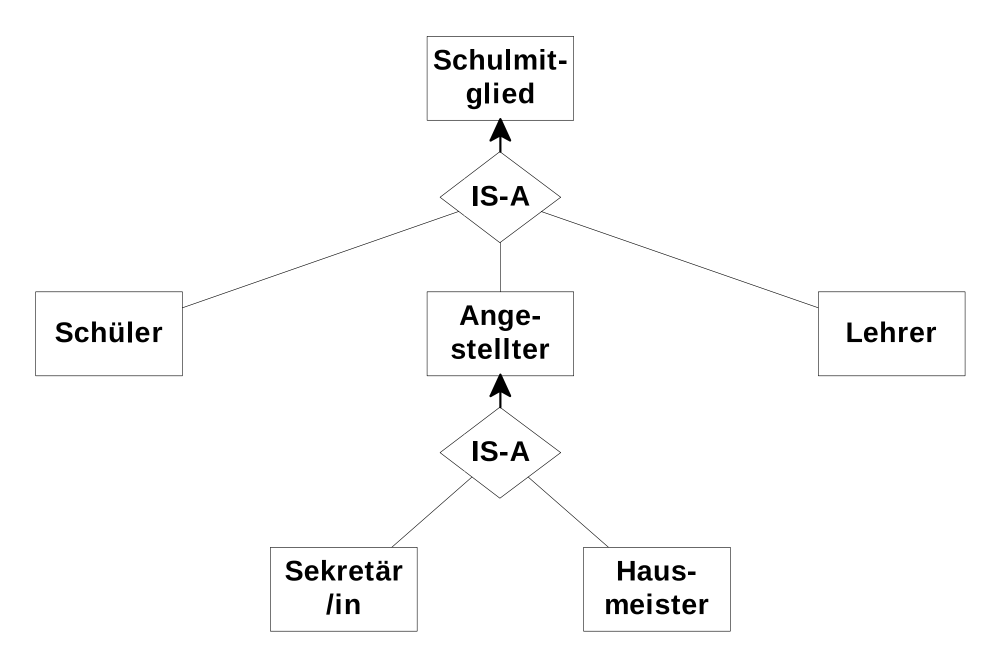


Abbildung: IS-A-Beziehung

Auf eine Angabe von Attributen wurde verzichtet. Es lassen sich jedoch ohne Probleme an diesem Beispiel ererbte und neue Attribute erläutern. So wären etwa "Name, Vorname, Adresse" für alle Schulmitglieder gemeinsam. Bei Lehrer könnten die Zusatzattribute „Titel, Unterrichtsfächer" auftauchen, bei Angestellter "Besoldungsgruppe" usw.

Abschließend noch ein etwas umfangreicheres Beispiel zum ER-Modell, das sich zum Teil aus den bisher besprochenen Beispielen zusammensetzt. Es beschreibt einen Ausschnitt aus der Schulverwaltung. Man kann sicher einige Kritikpunkte finden, da an eine komplette Modellierung der Miniwelt Schulverwaltung nicht gedacht ist.

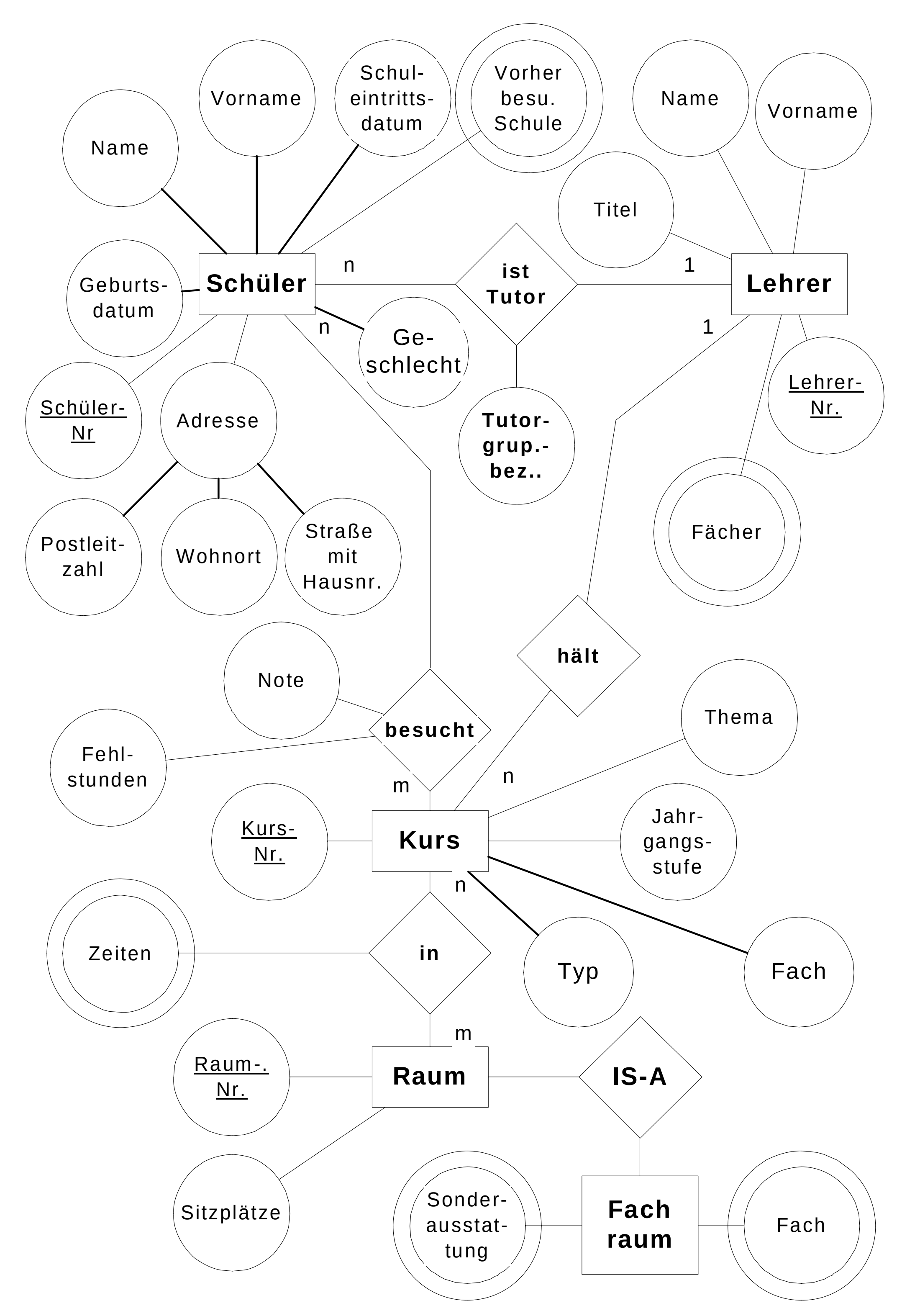


Abbildung: ER-Diagramm Schule

Aufgabe:
Beschreiben Sie das ER-Diagramm Schule verbal. Beschreiben Sie ferner jeden Beziehungstypen verbal, so wie es im Lehrtext gemacht wurde. Beispielsweise “Jeder Lehrer unterrichtet 1 Klasse. Jede Klasse wird von einem Lehrer unterrichtet“.

Lösung:

Im ER-Diagramm sind die wesentlichen Partizipanten einer Schulverwaltung erfasst. Es sind dies

- Räume
- Kurse
- Schüler
- Lehrer.

Als Besonderheit ist zu vermerken, dass jeder Schüler einen Lehrer als Tutor hat und Fachräume als besondere Ausprägung der Allgemeineren Entity 'Raum' modelliert wurden.

Jeder Schüler hat einen Lehrer als Tutor und besucht mehrere Kurse, die in verschiedenen Räumen stattfinden.
Jeder Lehrer ist für mehrere Schüler als Tutor zuständig und unterrichtet mehrere Kurse.
Die Attributierung der Entities ist intuitiv, z.B. gehören zu einem Schüler sein Name und eine Schülernummer, der zugleich als Primärschlüssel fungiert.

2.7 Die “Informatik-Bibliothek“ im ER-Modell

Das im letzten Kapitel vorgestellte ER-Modell soll jetzt konkret auf unser Projekt angewendet werden. Mit “ER-Modell“ ist zweierlei gemeint: Zum einen beschreibt es die Technik des Modellierens, d.h. der Abbildung der Realität. Zum anderen ist das Endprodukt gemeint, d.h. das abstrakte Modell, das in grafischer Form dargestellt werden kann.

Bei der Modellierung geht es zunächst darum, geeignete Entities zu definieren und mit notwendigen Attributen zu versehen. Beginnen wir zunächst mit den Ausleihern. Es sollten laut Anforderungsanalyse (vgl. Kapitel 2.5) *Name* und *Vorname* gespeichert werden. Als Schlüsselattribut führen wir *AusleiherNr.* ein, um allen Schwierigkeiten mit eventuell gleichen Vor- und Nachnamen aus dem Weg zu gehen. Die Zusatzattribute *Klasse* und *Tutor* bei Schülern erfassen wir durch den Sub-Entity-Typ *Schüler* des eigentlichen Typs *Ausleiher*, die durch eine IS-A-Beziehung verknüpft werden.

Etwas schwieriger ist schon die Modellierung der Bücher. Man muss sich zunächst klarmachen, dass es hier einen Unterschied zwischen dem Buch als Typ und den Exemplaren dieses Buchtyps gibt. Allgemeine Attribute wie *Autor*, *Titel* usw. charakterisieren eine bestimmten Typ von Büchern, von denen es natürlich mehrere Exemplare geben kann; z.B. bestimmt „Sedgewick: Algorithmen“ den Typ. Davon kann es aber zwei Exemplare im Informatikraum geben, von denen eins Präsenzbestand ist und nicht ausgeliehen werden soll. Schlüssel für die Exemplare ist die Inventarnummer. Dieser Unterschied lässt sich am besten erfassen durch eine IS-A-Beziehung zwischen den Entity-Typen *Buchtyp* und *Buch*.

Eine Anmerkung noch zum Attribut *Autor* von Buchtyp. Es erschien hier sinnvoll, ein Mehrfachattribut zu benutzen, denn häufig besitzen Bücher zwei oder mehr Autoren. Die könnte man natürlich als einen Wert des Attributs zusammenfassen, z.B. „Schlageter/Stucky“. Das würde jedoch Probleme bei der Suche nach dem Einzelautor mit sich bringen, denn wie findet man bei solchen zusammengesetzten Autorennamen noch „Stucky“.

(*Anmerkung: Im Kapitel über Normalformen lernen Sie die formale Analyse von Tabellen kennen Dabei werden Sie feststellen, dass Wert auf so genannte* atomare Attribute *gelegt wird, d.h. in jeder Zelle/bei jedem Attribut soll nur ein Wert stehen, also z.B. nur Schlageter statt Schlageter/Stucky).*

Kompliziert wird die Modellierung der Bücher außerdem noch durch das Erfassen von Büchern an anderen Standorten, denn für diese sollte nach Anforderungsanalyse keine unnötige Inventarnummer gespeichert werden. Es ist also nicht notwendig, jedes physische Exemplar zu erfassen. Es ist aber möglich, dass sich Exemplare desselben Typs an unterschiedlichen Standorten befinden (z.B. fünf Exemplare des Buchtyps „Informatik-Duden“ in der Bibliothek der gesamten Schule und drei weitere in der Bibliothek der kaufmännischen Abteilung). Das lässt sich gut durch einen weiteren Sub-Typ *Fremdbuch* von *Buchtyp* erfassen. *Fremdbuch* erhält kein eigenes Schlüsselattribut, denn jedes Entity ist eindeutig durch die IS-A-Beziehung und das Attribut *Ort* charakterisiert. Sinnvollerweise vergibt man für *Buchtyp* den Schlüssel *Buchtyp-Nr*.

Zentrale Beziehung für die Ausleihe ist die Beziehung *leiht_aus* zwischen den Entities *Ausleiher* und *Ausleihbuch*. Zu dieser Beziehung gehört als Attribut *Ausleihdatum*. Das Rückgabedatum muss nicht eigens gespeichert werden, es ergibt sich aus dem Ausleihdatum und dem zu Ausleihbuch gehörenden Attribut *Ausleihzeit*.

Jetzt bleibt nur noch die Erfassung der Stichwörter. Das stellt aber kein Problem dar. Mit Hilfe des Entity-Typs *Stichwort*, der als einziges Attribut *Stichwort* selbst hat, und der n:m -Beziehung *gehört_zu* mit den Entities von *Buchtyp* lässt sich ein Stichwortkatalog realisieren. Jetzt kann man das ER-Modell als Diagramm zusammenfassen.

Die Geschäftsregeln sind in unserem Modell sehr einfach gehalten:

(1) Die Objekte *Ausleiher* und *Ausleihbuch* existieren unabhängig voneinander. Ein Schüler kann Ausleiher sein, muss aber nicht.

(2) Jedes Ausleihbuch und jedes Fremdbuch gehören jeweils zu genau einem Buchtyp und existieren ohne diesen nicht.

(3) Zu jedem Buchtyp gehören ein oder mehrere Stichwörter. Stichwörter ohne zugehörigen Buchtyp existieren nicht.

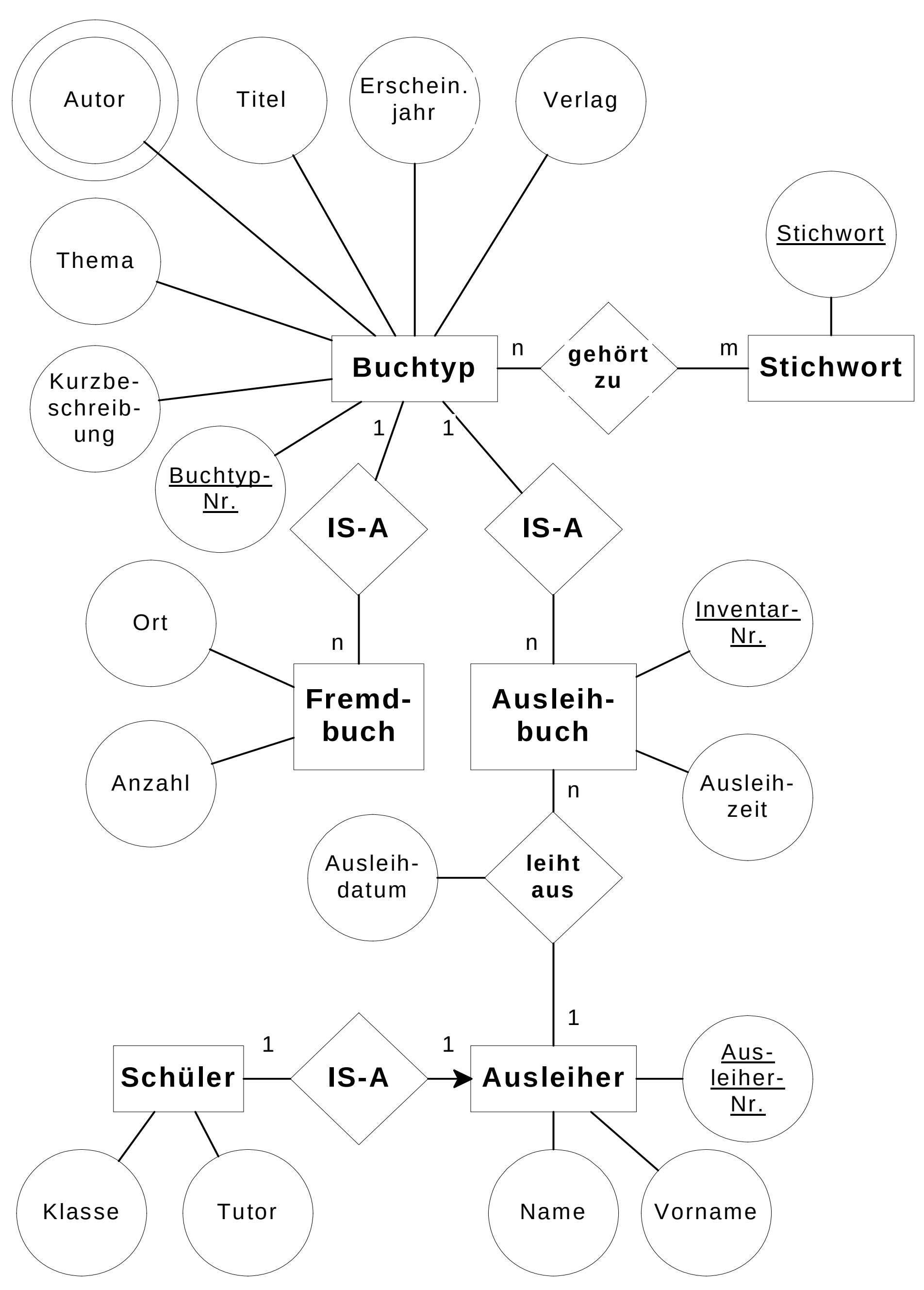


Abbildung: ER-Diagramm Informatik-Bibliothek

Eine kompakte Darstellung von Entitäten mit ihren Attributen verwendet die Mengendarstellung, wobei der Primärschlüssel unterstrichen wird. Bei einem zusammen gesetzten Primärschlüssel werden entsprechend die relevanten Felder unterstrichen. Allgemein: *Entityname{Primärschlüssel, Attribut 1, Attribut 2, …, Attribut n}*, z.B. Ausleihbuch{Inventar-Nr., Ausleihzeit}. Im Folgenden wird diese Notation *Mengendarstellung* genannt.

Aufgaben

Zur Modellierung von Ausschnitten der Wirklichkeit

Aufgabe 1:

Die folgenden Musterrechnungen dokumentieren eine Miniwelt “Rechnungschreiben“ in einer Firma.

Herrn
Horst Staniczek
Birnbaum 3
65510 Hünstetten

Rechnungsnummer:	R123	Rechnungsdatum:	11.06.1995
Kundennummer:	K002	Rechnungsbetrag:	1397,00

Position	Artikelnummer	Bezeichnung	Anzahl	Einzelpreis	Gesamtpreis
1	A3257	Minitor 17“	2	499,00	998,00
2	A4210	Nadeldrucker	1	399,00	399,00

Firma
Irma Computer GmbH&Co KG
Am Festungswall 45
65189 Wiesbaden

Rechnungsnummer:	R457	Rechnungsdatum:	04.03.1996
Kundennummer:	K195	Rechnungsbetrag:	359,00

Position	Artikelnummer	Bezeichnung	Anzahl	Einzelpreis	Gesamtpreis
1	A3117	CD-Rom	1	359,00	359,00

a) Stellen Sie fest, welche Objekte und Beziehungen sich daraus ableiten lassen und stellen Sie Ihre Ergebnisse in einer Tabelle der Form

Objekt | Beziehung | Eigenschaften

dar.

b) Formulieren Sie Geschäftsregeln für die Miniwelt.

c) Beschreiben Sie Objekte und Beziehungen durch ihre Eigenschaften und erstellen Sie entsprechend ein Modell der Miniwelt, indem Sie ein ER-Modell erstellen.

Aufgabe 2:

Alle im deutschen Sprachraum erscheinenden Bücher werden für die Deutsche Bibliothek nach einheitlichen Maßstäben gekennzeichnet. Die beiden folgenden Angaben stammen aus Büchern zu unserem Thema.

Moos, Alfred: SQL-Datenbanken: der Weg vom Konzept zur Realisierung in dBase; eine schrittweise und praxisnahme Einführung/ Alfred Moos; Gerhard Daues. - Braunschweig: Vieweg, 1991 ISBN 3-528-05183-3 NE: Daues, Gerhard (NE=Nebenautor)

Heuer, Andreas: Objektorientierte Datenbanken: Konzepte, Modelle, Systeme/ Andreas Heuer: - Bonn; München; Paris [u.a.]: Addison-Wesley, 1992 ISBN 3-89319-315-4

Autoren können dabei als Einzelautor, Nebenautor oder als Herausgebern fungieren. Ein Buch kann beim gleichen Verlag in mehrfachen Ausgaben oder bei verschiedenen Verlagen erscheinen.

a) Erstellen Sie ein Modell der Miniwelt „Bibliothek“, indem Sie die Entities *Autor, Buch* und *Verlag* sowie die Beziehungen *hat_Urheberschaft_als* für die Autorenschaft und *ist_Ausgabe_von* für die Verlagsausgabe verwenden!

b) Formulieren Sie entsprechende Geschäftsregeln, z.B. hat ein Buch mindestens einen Autor, ein Autor kann verschiedene Bücher schreiben.
Was wäre an Ihrem Modell, welches auf den obigen Daten basiert, noch zu verbessern, wenn es um die eindeutige Identifizierung von Datensätzen geht?

Aufgaben

Zur Komplexität von Beziehungen

Aufgabe 1:

Die folgende Tabelle zeigt jeweils zwei Entity-Typen und den zugehörigen Beziehungstyp. Geben Sie für jede Beziehung die Komplexität an.

(1)	Schüler	hat	Tutor
(2)	Schüler	bekommt heute	Zeugnis
(3)	Schüler	darf arbeiten an	Computer
(4)	Schüler	hat ausgeliehen	Buch
(5)	Schüler	besucht	Kurs
(6)	Schüler	ist befreundet mit	Schüler

Aufgabe 2:

Überlegen Sie, welche Komplexität die IS-A-Beziehung besitzt.

Aufgabe 3:

Gegeben sind jeweils zwei Entitytypen und ein Beziehungstyp. Geben Sie die jeweilige Komplexität an.

E-Typ 1	E-Typ 2	Beziehungs-Typ
Vater	Tochter	hat
Fuß	Zehe	gehört zu
Onkel	Neffe	hat
Schüler	Lehrer	hat Unterricht
Person	Personalausweis	besitzt
PROGRAM	FUNCTION	benötigt
Bruder	Schwester	hat
Ort	Ort	kürzeste Entfernung

Aufgabe 4:

Gegeben sind die Objekte
Klasse, Klasse, Klasse, Klassenleiter, Schüler, Lehrer
Und die Beziehungen
hat, enthält, unterrichtet.

Stellen Sie für je 2 Objekte je eine Beziehung her und geben Sie die Kardinalität an.

Aufgaben

Zum Entity-Relationship-Modell

Aufgabe 1:

Der folgende Ausschnitt aus einem Kursverzeichnis dient zur Kurswahl der Schülerinnen und Schüler zur Jahrgangsstufe 13. Ergänzen Sie mit den darin befindlichen Kursdaten unser Modell zur Kursverwaltung und stellen Sie fest, welche Entities bzw. Entity-Eigenschaften hinzugefügt werden müssen!

Leiste G6: Mi 9./10. Std. und Fr 5. Std

5 D 13	Kreß	165	Literatur und Gesellschaft
5 D 16	Biedermann	164	Literatur und Gesellschaft
5 E 12	Meyer K.	127	The Individual and Society
5 K 10	Blum	7	Kunst und Gesellschaft

Leiste G8: Do 1.2. Std. und FR 6. Std

5 D 10	Biedermann	170	Literatur und Gesellschaft
5 D 17	Beste	164	Angst
5 E 11	Meyenburg	125	The Individual and Society
5 K 11	Repert	12	Kunst und Gesellschaft
5 Mu 10	Lamprecht	140	Musik und Gesellschaft

Aufgabe 2:

Geben sie die Komplexität der folgenden Situationen an. Erstellen Sie jeweils ein ER-Diagramm. Formulieren Sie sinnvolle Geschäftsbedingungen.

a) Eine Ladenkette möchte Informationen über ihre Filialen und über deren Zulieferer speichern. Jeder Zulieferer beliefert mehrere Filialen, und jede Filiale kauft von mehreren Zulieferern.

b) Ein Installateur möchte über seine Handwerker und über die Häuser, in denen sie gerade arbeiten, Informationen speichern.

Aufgabe 3:

Die Computerzubehörfirma Microtec GmbH möchte ihre Verwaltung auf EDV umstellen. Sie verkauft ein Sortiment von Artikeln, die sie von verschiedenen Herstellern bezieht. Außerdem hat sie einen bestimmten Kundenkreis, der bei ihr Bestellungen aufgibt. Eine Bestellung kann natürlich mehrere Artikel umfassen. Derselbe Artikel kann oft von mehreren Herstellern bezogen werden, und ein Hersteller liefert natürlich meist mehr als einen Artikel.
Erstellen Sie im Entity-Relationship-Modell ein sinnvolles Datenmodell für die Firma, das Datenredundanz vermeidet. Wählen Sie geeignete Entities mit notwendigen Attributen und geben Sie die zwischen den Entities bestehenden Beziehungen mit ihrem Komplexitätsgrad an.

Aufgabe 4:

Zugrunde gelegt werde das ER-Modell der Schulverwaltung, wie es im Text aufgestellt wurde. Geben Sie bei den nachfolgenden Änderungen an, welche Ebene des DBMS von der Änderung betroffen ist.

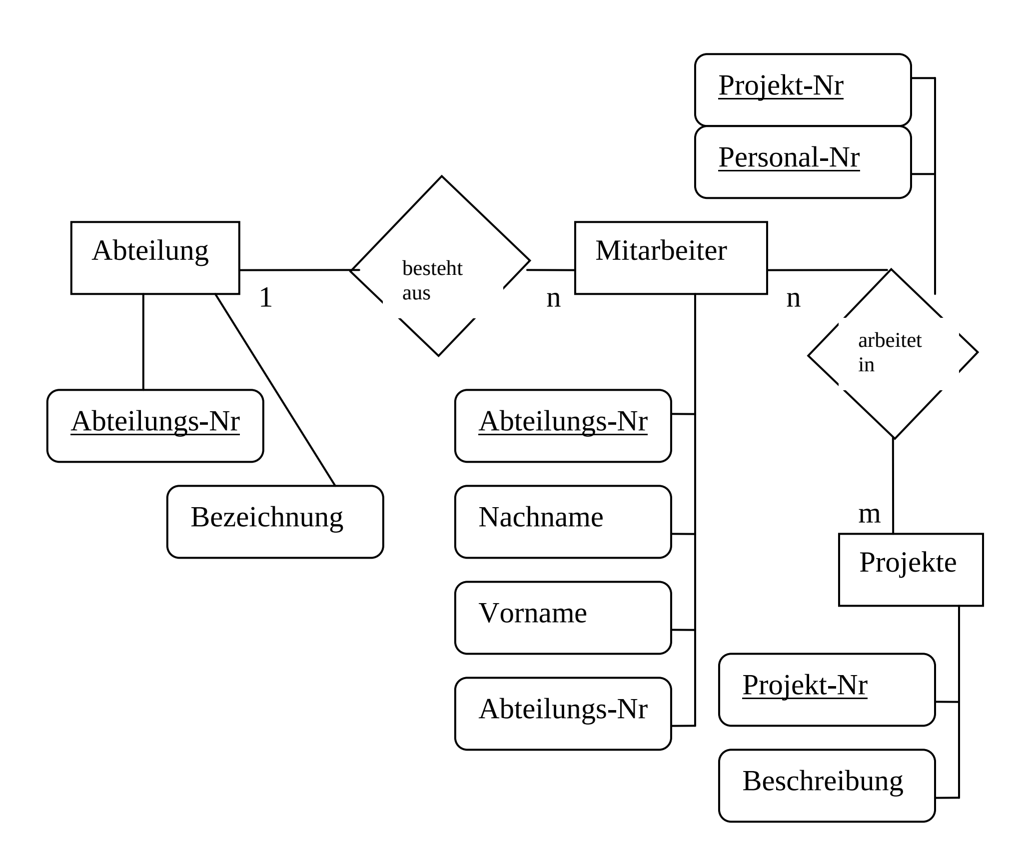
1. Der Lehrer Franz Schlauspruch kommt neu an die Schule.
2. Die Lehrer sollen bei der Noteneingabe nicht mehr die Noten des Schülers bei anderen Lehrern abfragen können.
3. Bei den Schülerdaten soll durch eine zusätzliche Indexdatei, in der die Schüler nach Wohnort sortiert sind, ein schnellere Suche nach Schülern eines Ortes ermöglicht werden, um die Busverbindungen besser koordinieren zu können.
4. Zur Erstellung von Alterstatistiken soll auch bei Lehrern das Geburtsdatum gespeichert werden.
5. Der Raum 556 soll als neuer Fachraum für Mathematik verwendet werden.
6. Auf Wunsch des Hausmeisters wird zusätzlich erfasst, welcher Lehrer einen Schlüssel für welchen Fachraum hat.
7. Neben dem/der Oberstufenleiter/in sollen auch die Tutoren die komplette Belegung ihrer Schüler einschließlich der bisher vergebenen Noten am Computer einsehen können.
8. Vom Sekretariat sollen die Schüler, die mehr als drei Grundkurse unter 5 Punkten einbringen müssen, per Serienbrief auf die Gefahr der Nichtzulassung zum Abitur hingewiesen werden.
9. Auf Antrag der SV dürfen Lehrer, die nicht Tutor des entsprechenden Schülers sind, die Fehlstundenzahl nicht mehr einsehen.
10. Daraufhin beschließt die Schulkonferenz, die Fehlstundenspeicherung ganz abzuschaffen.

Aufgabe 5:

Hugo Unbedarft besitzt eine große Spedition. Er will seine Auftragsverwaltung auf EDV umstellen und macht sich dazu einen genauen Plan. Seine Aufträge sind immer so, dass sie nur zu einem Ziel führen, es kann allerdings möglich sein, dass mehrere LKWs für einen Auftrag nötig sind. Nicht jeder LKW-Typ ist dazu geeignet, alle Ziele zu erreichen (z.B. zu niedrige Brücken), und nicht jeder Fahrer kann jeden LKW-Typ fahren. Hugo will folgende Daten speichern: AuftragsNr. und LKW-Nr., Ziel, Zielentfernung, Auftragsdatum, LKW-Typ, max. Zuladung eines LKW-Typs, TÜV-Datum, Fahrer-Nr, Fahrer-Name.
Erstellen Sie ein ER-Modell für die Spedition.

Aufgabe 6:

Gegeben ist folgendes ERM:



a) Leiten Sie daraus die kompakte ERM-Notation ab: Tabellenname(Primärschlüssel, Attribute)

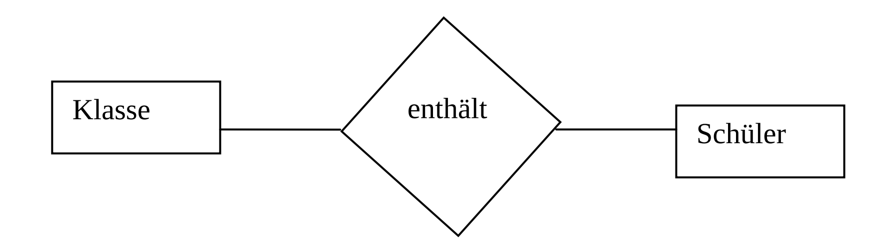
b) Geben Sie die Schritte an, wie aus einem ERM ein relationales Modell wird.

Aufgabe 7:

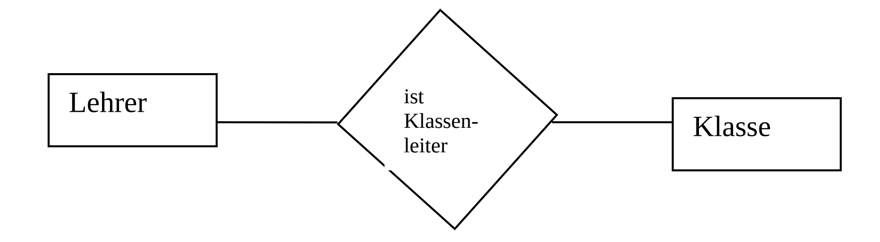
Wir bauen nun sukzessive ein vereinfachtes Lehrer-Schüler Unterrichtsmodell auf.

a) Bestimmen Sie die beteiligten Objekte.
b) Ermitteln Sie für alle Schritte die Kardinalitäten.
c) Leiten Sie nun einmal versuchsweise das ERM her. Details folgen im nächsten Kapitel. Versehen Sie dabei die Objekte mit sinnvollen Attributen.

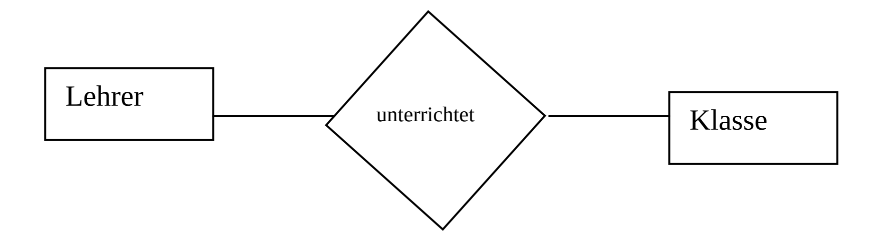
Schritt 1:



Schritt 2:



Schritt 3:



Lösungsheft

Kapitel 2
–
Das konzeptuelle Modell

Kapitel: Datenbanken – Mehr als Tabellen? Wozu das Ganze?

Aufgabe:

a) Welche Nachteile hat eine Datenerfassung dieser Art?



b) Nennen Sie andere Bereiche, in denen Datenbanken eingesetzt werden.

Lösung:

a)

Die Probleme, die hierbei auftreten, sind ersichtlich:

- **Datenredundanz**, d.h. dieselben Daten werden mehrfach gespeichert.

- Gefahr der **Dateninkonsistenz**, d.h. bei Änderungen können wegen der mehrfach gespeicherten Daten Widersprüche auftreten. Verlässt ein Schüler die Schule, so wird er mit dem Programm im Sekretariat gelöscht, aber mit einem anderen immer noch in Kurse eingeteilt (es soll sogar schon vorgekommen sein, dass solche fiktiven Schüler auch Noten erhalten haben).

- **Unflexibel** gegenüber Änderungen, kurz gesagt, Änderungen der Datenstruktur verlangen Änderungen der Programme; zusätzliche Aufgaben verlangen neue Datenorganisation, da diese an der alten Zielsetzung ausgerichtet waren. Abfragen über mehrere Datenbestände hinweg lassen sich in diesem System kaum realisieren, (z.B. eine Abfrage über die Daten der Bibliothek und der Kurseinteilung: „Lesen Schüler der Deutsch-Leistungskurse mehr klassische Literatur?“.

- **Datenschutzprobleme** und Datensicherheit, Benutzer bzw. Programme besitzen vollen Zugriff auf die dem Programm zugeordneten Daten; so könnte z.B. das Bibliotheksprogramm aufgrund einer Fehlfunktion sämtliche Ausleiher, d.h. Schülerdaten, löschen.

- **Keine** Einhaltung von **Standards**, d.h. Datenbestände werden an unterschiedlichen Stellen in unterschiedlichen Formaten abgespeichert und sind nicht ohne weiteres austauschbar; z.B. können die Schülerdaten im Sekretariat in einem Format vorliegen, das vom Bibliotheks-programm nicht verstanden wird, so dass diese Daten dort extra eingegeben werden müssen.

b)

Schülerverwaltung
Personalverwaltung
Kunden-/Auftragserfassung
x-Sammlung (CD, Videos,…)
für Statistiken (als Datenquelle)
Bibliothek
Bank
Benutzerverwaltung
im Internet

- Standardpaket XAMPP:
  Apache Webserver, Linux oder Windows Betriebssytem, PHP (Programmiersprache), MySQL (Datenbank)
- DB ist **Backend**, als **Frontend** fungiert eine Browserapplikation
- Basis für *dynamische Webseiten* (z.B. Begrüßungen, Uhrzeit, …)
  in modernen Betriebssytemen
  (Active Directory unter Windows Server ist letztlich eine Datenbank)

Aufgabe:

a) Wie würde eine ,,Datenbank“ mit einer Tabellenkalkulation aussehen? Machen Sie sich das am besten an Hand eines konkreten Beispiels klar. Um den Überblick über die Ausgaben jedes Klassenmitglieds zu behalten, beschließen Sie, die Einkaufdaten in einer ,,Excel-Datenbank“ zu verwalten. Überlegen Sie sich, welche Daten Sie benötigen.

b) Öffnen Sie nun die Excel-Datei “*Kassenbon.xls*“ und schauen Sie sich die dortigen Vorschläge an. Vergleichen Sie Ihre Ideen mit denen in der Datei, sie müssen nicht schlechter sein!

c) Nehmen Sie dann einige Einfüge-/Lösch- und Suchoperationen vor. Was fällt Ihnen auf?

d) Welche Nachteile gibt es bei der Datenverwaltung mit einem Tabellenkalkulationsprogramm? Belegen Sie Ihre Vorschläge an Hand des konkreten Beispiels.

Lösung:

	A	B	C	D	E	F	G	H	I	J	K	L
1	SchülerIn	Kl.	w/m	Sparte	Artikel	Preis	Zahlung	MWST	Geschäft PLZ Ort Tel	Datum	Zeit	Besonderes
2	joel brandeis	2f00	m	Lebensmittel	Eier	SFr. 2,80	bar	231102	Migros Höngg	06.04.00	15:30	Kasse 1
3	joel brandeis	2f00	m	Lebensmittel	Sprite	SFr. 1,20	bar	380507	Epa 8001 Zürich	02.02.01	12:24	
4	joel brandeis	2f00	m	Lebensmittel	Käse	SFr. 4,85	bar	231074	Coop Zürich-Linth	26.06.95	12:39	
5	Alana Gerdes	2b01	w	Transport	Tageskarte Zone 10 21 22 30 31 3	SFr. 5,20	bar		Looren Apotheke Witikonerstrasse 397	01.09.99	3:36	0230 0001 00004006 00142
6	Alana Gerdes	2b01	w	Lebensmittel	Farmer-Stengel	SFr. 4,00	Bar	388507	Epa Aarau	11.04.00	10:03	Kasse 3
7	Alana Gerdes	2b01	w	Kleider	Nachtwollhemd	SFr. 30,00	Bar	231102	Migros Ebmatingen	22.03.01	9:22	
8	Caroline Diethelm	2b01	w	Unterhaltung	Notting Hill	SFr. 11,00		231074	Coop Zürich-Linth	04.09.99	13:50	Balkon links Grosse Bühne Reihe 10 Platz 775
9	Caroline Diethelm	2b01	w	Lebensmittel	Rohschinken	SFr. 15,42	bar	231074	coop Dietikon 8953	01.09.00	16:19	Fr. B. Holzer Kasse 01
10	Caroline Diethelm	2b01	w	Transport	Tageskarte	SFr. 5,20	bar	231102	Migros Rigiplatz 8006 Zürich	02.09.00	9:35	Frau B. Holzer Kasse 1
11	Caroline Diethelm	2b01	w	Transport	Einzelbillett	SFr. 2,60	Karte	231102	Migros Rigiplatz	21.03.01	9:29	Frau N.kuhn
12	Caroline Diethelm	2b01	w	Lebensmittel	Truffes-Cake gross	SFr. 27,00	Bar	231102	M Rigiplatz	03.09.99		Kasse 2
13	Caroline Diethelm	2b01	w	Papeterie	Papeterie	SFr. 1,40	Karte	231074	Coop Zürichbergstr.	05.08.00	11:36	M.Mueller
14	Caroline Diethelm	2b01	w	Lebensmittel	Barclay	SFr. 4,70	bar	231074	Coop Zürichbergstrasse 8032 Zürich	17.03.01	13:39	R.Budel, Kasse 1
15	Caroline Diethelm	2b01	w	Lebensmittel	Toni Vollrahm	SFr. 2,10	Bar	231074	Coop Zuerich	03.03.00	15:55	
16	Caroline Diethelm	2b01	w	Lebensmittel	Tragtasche	SFr. 0,30	bar	235289	Studentenladen	25.07.00	12:54	
17	Caroline Diethelm	2b01	w	Lebensmittel	Paprikachips	SFr. 3,95	bar	231074	COOP Zuerich-Linth ,Cl. Römerhof	07.04.00	7:40	
18	Caroline Diethelm	2b01	w	Lebensmittel	Zwiebeln gehackt	SFr. 1,90	Bar	251030	Body Shop 8001 Zürich 2606922	10.12.99	15:02	
19	Caroline Diethelm	2b01	w	Lebensmittel	DIM SUM Poulet	SFr. 4,50	Karte		Selecta Automat	10.10.00	14:50	
20	Caroline Diethelm	2b01	w	Lebensmittel	Edelnuss Misch.	SFr. 3,90	bar	231074	Coop Zürich-Linth St. Annahof Volkilar	10.03.00	17:05	
21	Caroline Diethelm	2b01	w	Lebensmittel	Toni Vm Past	SFr. 1,60	Bar	231074	Coop 8706 Feldmeilen	07.09.00	12:31	
22	Caroline Diethelm	2b01	w	Lebensmittel	Nektarinen Gelb	SFr. 0,50	bar	231102	Migros Höschgasse Zürich	31.08.00	14:40	
23	Caroline Diethelm	2b01	w	Papeterie	Mäppchen	SFr. 1,90	bar	231124	Beach Mountain	24.07.00	15:16	
24	Caroline Diethelm	2b01	w	Lebensmittel	Melonen	SFr. 1,80	Bar	380507	EPA Bellevue Zürich	03.04.00	17:57	
25	Caroline Diethelm	2b01	w	Kleider	Hose Figura	SFr. 6,50	Bar	231074	COOP Zürich-Linth	17.07.99	10:12	
26	Fabian Weber	2e99	m	Papeterie	Karten	SFr. 4,20	Bar	231074	Coop Zürich-Linth CL Römerhof	01.02.01	14:17	2
27	Fabian Weber	2e99	m	Papeterie	papeterie	SFr. 2,50	Bar	230820	Media Markt Dietlikon 01/805 95 95	31.08.00	20:30	2
28	Fabian Weber	2e99	m	Papeterie	Servietten	SFr. 3,00	bar		Kino ABC	22.08.00	9:36	001 0103
29	Fabian Weber	2e99	m	Übriges	Servietten	SFr. 1,70	bar	175287	Pickpay Discount 8703 Erlenbach 40/4	26.11.99		6 Aushilfe
30	Fabian Weber	2e99	m	Lebensmittel	Nudeln	SFr. 2,00	Bar	231074	Coop Zürichbergstrasse 8032 Zürich	22.03.00	15:04	Cherryl
31	Fabian Weber	2e99	m	Kleider	Bluse	SFr. 79,00	Bar	231102	Erlibacher Märt 8703	08.04.00	18:06	Kasse 1

Mehrfacherfassungen

leere Felder

!= Schreibweisen

Daten des Käufers müssen für jeden Kauf neu eingeben werden

→ Datenredundanz

→ Was wenn sich mal die Adresse ändert? Änderungen wären an vielen Stellen mit hoher Fehleranfälligkeit zu machen

Es kann unterschiedliche Schreibweisen für ein und dieselbe Sache geben

→ Probleme beim Suchen

→ Dateninkonsistenz

Es gibt viele leere Felder

→ Warum nicht gleich weglassen? Geht ja nicht, wegen den Ausnahmen. Wäre aber schön

Eingabemaske ist eher unflexibel

Prima, wir haben jetzt die Situation durchdrungen. Bearbeiten Sie die folgenden Aufgaben der Reihe nach.

Aufgabe 1:



Was müsste prinzipiell geändert werden, um die identifizierten Probleme zu beseitigen?

Aufgabe 2:

Auch Access selbst kann Tabellenstrukturen analysieren. Mal sehen, ob die Vorschläge des Programms Ihren Überlegungen ähnlich sind. Starten Sie Access, legen Sie eine leere neue Datenbank an (Datei→Neu→leere Datenbank) und importieren Sie über Datei→externe Daten→importieren→Kassenbondatenbank.xls auswählen→Fertig stellen. Schauen Sie sich die importierten Daten an und rufen dann den Analyseassistenten auf: Extras→Analyse→Tabelle. Verfolgen Sie nun die Vorschläge des Assistenten und bringen Sie sie mit Ihren Überlegungen und der bisher erarbeiteten Problematik in Deckung. Ergänzen Sie gegebenenfalls Ihre Überlegungen.

Aufgabe 3:

Öffnen Sie die Datei "ExcelDatensammlung". Hier wurden einfach alle Daten einer Einkaufsliste und einer statistischen Datenerhebung in einer Tabelle erfasst. Machen Sie sich Gedanken, wie Sie die Daten sinnvoll in verschiedene Tabellen aufteilen können. Welche Kriterien wenden Sie an? Und wie machen Sie deutlich, wie die einzelnen Tabellen in Verbindung stehen?

Anm.: Die Anzahl an Spalten und Tabellen in der leeren Vorlage muss mit Ihrer Lösungsidee nicht zwangsläufig übereinstimmen. Zeichnen Sie gegebenenfalls einfach neue Spalten und Tabellen hinzu.

Lösung:

- Trennung von Stamm- und Bewegungsdaten. Stammdaten sind die Basisdaten, z.B. Adressen mit Namen, die zentral vorgehalten werden, während Bewegungsdaten zusätzliche Daten darstellen, die zu den Stammdaten gehören und durch einen Sachverhalt (mehrfach) erzeugt werden, z.B. die konkreten Einkaufsvorgänge usw.

- Aufteilen in mehrere Tabellen → Problem wird Zergliedert

Beispiele:

Beispiel 1: Kursbesucher

Name	Wohnort	Telefonnr.	Datum Kursbesuch	Bezahlt?
Urs Frey	Wald	786 15 12	21. Januar	Ja
Urs Frey	Wald	786 15 12	28. Februar	Nein
Angie Becker	Uster	755 15 39	21. Januar	Ja

Datum Kursbesuch	Bezahlt?	Name
21. Januar	Ja	Urs Frey
28. Februar	Nein	Urs Frey
21. Januar	Ja	Angie Becker

Name	Wohnort	Telefonnr.
Urs Frey	Wald	786 15 12
Angie Becker	Uster	755 15 39

Beispiel 2: Kursdatenbank

Datum	Kursname	Kursinhalt	Preis
21. Januar	Start 1	Einführungskurs	120 Fr.
28. Februar	Start 2	Fortgeschritten	100 Fr.
19. März	Start 1	Einführungskurs	120 Fr.

Datum	Kursname
21. Januar	Start 1
28. Februar	Start 2
19. März	Start 1

Kursname	Kursinhalt	Preis
Start 1	Einführungskurs	120 Fr.
Start 2	Fortgeschritten	100 Fr.

Der Analyseassistent stellt genau das fest. Es gibt redundant gespeicherte Daten, was Platz kostet, Fehler fördert und ineffizient ist. Er schlägt dann Lösungen vor, die allerdings mit Vorsicht zu genießen sind.

Lösung zu “ExcelDatensammlung“ in „ExcelDatensammlung_LSG“.

Kapitel: Ebenen eines Datenbankmanagement-Systems

Aufgabe:

Welche Anforderungen werden – vor dem Hintergrund des just erworbenen Wissens – also an eine Datenbank und ihr Managementsystem (DBMS) gestellt? Denken Sie auch darüber hinaus. Was passiert bei mehreren Benutzern, wie werden Daten gesichert usw.

Lösung:

Eine Datenbank (DB) speichert Daten. Dabei sind verschiedenen Aspekte zu beachten, die die

- physikalische Speicherung (physische Ebene/interne Sicht), ihre
- Organisation (logische Ebene/konzeptuelle Sicht), ihre
- Auswertung (Anfragen) und ihre
- Darstellung / Präsentation (Sichten)

betreffen.

Faktoren die zu leisten sind:

- Informationsspeicherung (potentiell große Mengen)
- Effektive Informationsverwaltung große Datenmengen:
    - schnelles Wieder finden und Zugreifen
- Ausnutzung der physikalischen Speichermedien
- Sicherung der Datenbestände (Backup und Restauration)
- Datensicherheit (Zugriff, versehentliche Manipulation, (Security)Updates, Restauration)
- Vermeidung von Redundanzen
- Vermeidung von Abhängigkeiten der Daten von anderen Programmen
- Datenintegrität: Korrektheit, Widerspruchsfreiheit, Konsistenz und Vollständigkeit der Daten
- Vermeidung von Anomalien (unerwünschtes Verhalten bei Änderungen der Daten)
- Leichte Wartbarkeit, gute Verständlichkeit
- Leichte Erweiterbarkeit
- Eventuell verteilte Datenhaltung
- Höchste Verfügbarkeit
- Mehrfachzugriff
- Unterstützung bei der Datenmanipulation (Ändern, Löschen, Einfügen)
- Hohe Zugriffszeit (Wichtigste: die durchschnittliche/mittlere Zugriffszeit)
- Realisierung der Sichten (Views)

Diese Funktionen sind in einer Datenbank umgesetzt. Sie besteht aus der eigentlichen
- Datenbasis,
also den tatsächlichen Daten, und einer Reihe von Verwaltungsprogrammen, dem
- **DBMS** (database management system).

Aufgaben

3-Ebenen-Architektur von Datenbanksystemen

Aufgabe 1:
Es soll eine Datenbank für eine Bücherei eingerichtet werden. Ein einzelnes Buch sei beschrieben durch die folgenden Merkmale:
Inventarnummer des Buches, ISBN-Nummer, Autor, Titel, Fachgebiet, Verlag, Erscheinungsort und -jahr, Auflage, Preis.

a) Zu welcher Ebene gehört diese Beschreibung?

Additum:
Nehmen wir an, die Datenbank soll in Pascal/Java/… geschrieben werden. Welche Hilfsmittel stehen dann zur Verfügung, um möglichst saubere Schnittstellen zwischen den einzelnen Ebenen zu realisieren? Welche Design-Entscheidungen bei der Planung des Programms gehören zur physischen, welche zur konzeptuellen, welche zur externen Ebene? Inwieweit lässt sich Datenunabhängigkeit wahren?

Aufgabe 2:
Analysieren Sie bei jeder der folgenden Umstellungen, auf welcher der Ebenen einer bestehenden Datenbank und an welchen Transformationen jeweils etwas geändert werden muss (im Idealfall):

a) Neuer Rechner, aufwärts kompatibel, gleiches Betriebssystem, gleiches Datenbanksystem
b) Neuer Rechner, anderes Fabrikat, gleiches Betriebssystem, gleiches Datenbanksystem.
c) Neue Platten mit anderer Struktur
d) Ein neues Programm benutzt bestehende Daten.
e) Ein neues Programm X benutzt bestehende Daten und zusätzliche neue Datenstrukturen.
f) Eine neue Verbindung zwischen zwei existierenden Typen wird eingeführt, bisher gab es z.B. zwischen Schülern und Lehrern nur die Verbindung, wer Tutor ist. Jetzt gebe es zusätzlich die Verbindung, wer für jeden Schüler frei wählbarer Vertrauenslehrer ist.
g) Der Bestand einer Datenbank wird geteilt.

Lösungen

3-Ebenen-Architektur von Datenbanksystemen

Aufgabe 1:
Es soll eine Datenbank für eine Bücherei eingerichtet werden. Ein einzelnes Buch sei beschrieben durch die folgenden Merkmale:
Inventarnummer des Buches, ISBN-Nummer, Autor, Titel, Fachgebiet, Verlag, Erscheinungsort und -jahr, Auflage, Preis.

a) Zu welcher Ebene gehört diese Beschreibung?

Additum: Für Delphi:
Nehmen wir an, die Datenbank soll in Pascal geschrieben werden. Welche Hilfsmittel stehen dann zur Verfügung, um möglichst saubere Schnittstellen zwischen den einzelnen Ebenen zu realisieren? Welche Design-Entscheidungen bei der Planung des Programms gehören zur physischen, welche zur konzeptuellen, welche zur externen Ebene? Inwieweit lässt sich Datenunabhängigkeit wahren?

a) Die Beschreibung gehört zum **konzeptuellen Schema**.

b) Eine saubere Schnittstelle zwischen dem konzeptuellen Schema und dem internen Schema lässt sich z.B. durch die logische Struktur des **ADT**, z.B. realisiert in Units, konstruieren. Hierdurch kann z.B. verborgen werden, ob intern ein Feld, eine lineare Liste und/oder eine Datei zur Verwaltung des Datenbestandes verwendet wird. Diese Entscheidungen gehören zum internen Schema. Die physische Datenunabhängigkeit läßt sich also recht gut realisieren. Die Wahl des Records und allgemeiner Methoden hierzu gehört zum konzeptuellen Schema. Die Realisierung einer externen Sicht ist nicht ganz so einfach. Sollen z.B "Informatiker" nur auf Bücher mit dem Fachgebiet "Informatik" zugreifen können, so müsste die allgemeine Zugriffsmethode bereits so implementiert sein, dass hierbei die Fachgebiete abgefragt werden. Auf solch eine Methode könnte man dann eine Methode zum "Ausblenden" aufsetzen. Die logische Datenunabhängigkeit lässt sich nur dann realisieren, wenn man die Methoden auf der konzeptuellen Ebene sehr breit (mit vielen Fallunterscheidungen) angelegt hat.

Aufgabe 2:
Analysieren Sie bei jeder der folgenden Umstellungen, auf welcher der Ebenen einer bestehenden Datenbank und an welchen Transformationen jeweils etwas geändert werden muss (im Idealfall):

a) Neuer Rechner, aufwärts kompatibel, gleiches Betriebssystem, gleiches Datenbanksystem

auf keiner Ebene Änderungen erforderlich (wirklich ideal!!!)

b) Neuer Rechner, anderes Fabrikat, gleiches Betriebssystem, gleiches Datenbanksystem.

interne Sicht: Änderungen lassen sich selbst bei gleichen Betriebssystemen nicht vermeiden, wenn die Hardware unterschiedlich ist (z.B. von VAX auf SUN). Die logische Gesamtsicht bleibt zwar erhalten, muss aber neu kompiliert werden (Änderung der Transformation). Die externe Sicht bleibt erhalten. Die Daten müssen während des Kompilierens ausgelagert werden, da die Rechner sicherlich auf Bitebene nicht kompatibel sind.

c) Neue Platten mit anderer Struktur
Da die physischen Datenstrukturen jetzt neu sind, sind Änderungen der internen Sicht notwendig. Keine Änderungen auf höheren Ebenen. Änderung der Transformation zwichen interner und logischer Sicht.

d) Ein neues Programm benutzt bestehende Daten.
Für Programm X muß ein neue externe Sicht erstellt werden. Auf den anderen Ebenen keine Änderungen. Erstellen einer Transformation für Programm X.

e) Ein neues Programm X benutzt bestehende Daten und zusätzliche neue Datenstrukturen.
Bei neuen Datentypen muss die interne Sicht geändert werden. Auch die logische Gesamtsicht muss u.U. zur Einbindung der neuen Strukturen geändert werden. Dies ist hoffentlich so zu erreichen, dass eine externe Sicht nur dann geändert werden muss, wenn die neuen Datentypen auch mit genutzt werden, z.B. für das Programm X, und nicht für die bisherigen Sichten.

f) Eine neue Verbindung zwischen zwei existierenden Typen wird eingeführt, bisher gab es z.B. zwischen Schülern und Lehrern nur die Verbindung, wer Tutor ist. Jetzt gebe es zusätzlich die Verbindung, wer für jeden Schüler frei wählbarer Vertrauenslehrer ist.

Da existierende Typen (Schüler/Lehrer)verbunden werden, ist eine Änderung der logischen Gesamtsicht notwendig. Da neue Informationen gespeichert werden müssen, ist eine Änderung der internen Sicht erforderlich.. Externe Sichten müssen nur geändert werden, wenn sie die neue Verbindung benutzen.

g) Der Bestand einer Datenbank wird geteilt.

Wenn man Daten gleichen Typs in zwei Mengen teilt, und die Programme getrennt auf diesen Mengen arbeiten sollen, so sind keine Änderungen erforderlich. Natürlich erhält man dann nur Auskunft über die Daten, die in diesem Teil der Datenbank gespeichert sind. Wenn man aber so teilt, daß z.B. zu einer Schülernummer die Daten wie Adresse u.ä. in der einen Menge und die Kurswahlen und Noten in der anderen Menge sind, so ändern sich alle Sichten; auf der externen Ebene ändern sich nur solche Anwendungen nicht, die gerade eine der abgeteilten Datenmengen nutzen.

Kapitel: Von der Realität zum Modell

Aufgabe 1:
Was verstehen Sie generell unter dem Begriff der 'Modellbildung' in der Informatik/Technik?

Aufgabe 2:
Erläutern Sie die Begriffe Objekt, Entity, Eigenschaften, Attribute und Beziehungen.



Aufgabe 3:
Wovon geht man zu Beginn der Modellbildung i.d.R. aus?



Aufgabe 4:
Nehmen Sie sich eine Mini-Welt Ihrer Wahl vor und überlegen Sie sich geeignete Inhalte für die Struktur.



Entity	Beziehung	Eigenschaften
…	…	…
…	…	…

Aufgabe 5:
Sie bekommen von einem Stahlwerk den Auftrag, den Schmelzvorgang für besondere Stahlsorten datenbanktechnisch zu erfassen. Beschreiben Sie generell und an Hand der kennen gelernten Fachbegriffe und Vorgehensmodelle, wie Sie an das Problem herangehen.

Lösungen

Aufgabe 1:
Was verstehen Sie generell unter dem Begriff der 'Modellbildung' in der Informatik/Technik?

Modellbildung ist der Abbildungsprozess der (hochkomplexen) Wirklichkeit auf die informationstechnisch relevante (vereinfachte) Wirklichkeit. Dabei ist es die Kunst zu ermitteln, was denn genau relevant ist. Dabei gilt es alles zu erfassen was wesentlich ist und ausreicht, um die gesetzten Ziele mit dem Modell auch erreichen zu können. Damit sind **Modelle immer Zielabhängig**! Ein Modell besteht aus Objekten mit ihren Eigenschaften und den Beziehungen untereinander.

Aufgabe 2:
Erläutern Sie die Begriffe Objekt, Entity, Eigenschaften, Attribute und Beziehungen.

Objekte sind unterscheidbare Einheiten, die gewisse Eigenschaften=Attribute besitzen und in Beziehung zu anderen Teilnehmern (Objekten) stehen. Im Bereich der DB spricht man statt von Objekten von Entities.

Aufgabe 3:
Wovon geht man zu Beginn der Modellbildung i.d.R. aus?

Man verschafft sich einen intensiven Eindruck vom abzubildenden Prozess/von der abzubildenden Situation, in dem man vorhandenes Material (z.B. Zeugnisse) sichtet und von der Quantität auf die allgemeine Struktur schließt. Dabei ist der Aspekt der 'Datensammlung' zu sehen, ebenso wie der Aspekt der Entscheidung was von den gesammelten Daten und Eindrücken im Zuge der Modellbildung weggelassen werden kann.

Aufgabe 4:
Nehmen Sie sich eine Mini-Welt Ihrer Wahl vor und überlegen Sie sich geeignete Inhalte für die Struktur.

Mini-Welt Zoo

Entity	**Beziehung**	**Eigenschaften**
Pfleger		Nachname, Vorname, Adresse, Einstellungsdatum, Gehaltsstufe, Pflegernummer
	pflegt	
Tier		Name, Rasse, Unterart, Rassennummer, Unterartennummer, Geburtsdatum, imZooSeit, Pflegernummer, Anamnese (Krankengeschichte), Nachkommen, Kommentar
	wohnt in	
Käfig		Käfignummer, Größe, Rassennummern

Was ist mit den Besuchern? Und den Eintrittspreisen? Nun, ist alles noch nicht erfasst.

Aufgabe 5:
Sie bekommen von einem Stahlwerk den Auftrag, den Schmelzvorgang für besondere Stahlsorten datenbanktechnisch zu erfassen. Beschreiben Sie generell und an Hand der kennen gelernten Fachbegriffe und Vorgehensmodelle, wie Sie an das Problem herangehen.

Als Informatiker hat man zunächst einmal überhaupt keine Ahnung wie so ein Schmelzvorgang abläuft. Also gilt es zunächst **'Prozessverständnis'** aufzubauen, in dem man sich vor Ort schlau macht (schauen, analysieren, lesen, diskutieren, fragen).
Hat man ein hinreichendes Verständnis der wesentlichen Abläufe erlangt gilt es die Größen zu ermitteln, die für das Erreichen des Zieles relevant sind, denn darauf ist jede Modellbildung ausgelegt. Diese Ebene der Analyse des Prozesses und Bewertung der Frage was vom komplexen Ablauf nun unbedingt relevant für das Modell ist und was nicht ist die kognitive Leistung, die es stets in **Zusammenarbeit mit den Fachexperten** zu bewältigen gilt!
Daraus erwächst ein erster Modell-Prototyp, der intensiv mit den Kollegen und den Fachexperten diskutiert wird. Ist sichergestellt, dass mit dem Modell die Ziele erreicht werden können, kann an der Detailstruktur gearbeitet werden. Ab hier iteriert der Zyklus von Modellbildung – Rücksprache – Redesign bis zum „fertigen“ Modell.
Letzteres kann dann konkret in ein Produkt (Datenbank + evtl. noch weitere Komponenten) umgesetzt werden.

Kapitel: Das Entity-Relationship-Modell

Aufgabe:
Beschreiben Sie das ER-Diagramm Schule verbal. Beschreiben Sie ferner jeden Beziehungstypen verbal, so wie es im Lehrtext gemacht wurde. Beispielsweise “Jeder Lehrer unterrichtet 1 Klasse. Jede Klasse wird von einem Lehrer unterrichtet“.

Lösung:

Im ER-Diagramm sind die wesentlichen Partizipanten einer Schulverwaltung erfasst. Es sind dies

- Räume
- Kurse
- Schüler
- Lehrer.

Als Besonderheit ist zu vermerken, dass jeder Schüler einen Lehrer als Tutor hat und Fachräume als besondere Ausprägung der Allgemeineren Entity ’Raum’ modelliert wurden.

Jeder Schüler hat einen Lehrer als Tutor und besucht mehrere Kurse, die in verschiedenen Räumen stattfinden.
Jeder Lehrer ist für mehrere Schüler als Tutor zuständig und unterrichtet mehrere Kurse.
Die Attributierung der Entities ist intuitiv, z.B. gehören zu einem Schüler sein Name und eine Schülernummer, der zugleich als Primärschlüssel fungiert.

Kapitel: Die “Informatik-Bibliothek“ im ER-Modell und Sammelaufgaben

Aufgaben

Zur Modellierung von Ausschnitten der Wirklichkeit

Aufgabe 1:

Die folgenden Musterrechnungen dokumentieren eine Miniwelt “Rechnungschreiben“ in einer Firma.

Herrn
Horst Staniczek
Birnbaum 3
65510 Hünstetten

Rechnungsnummer:	R123	Rechnungsdatum:	11.06.1995
Kundennummer:	K002	Rechnungsbetrag:	1397,00

Position	Artikelnummer	Bezeichnung	Anzahl	Einzelpreis	Gesamtpreis
1	A3257	Minitor 17“	2	499,00	998,00
2	A4210	Nadeldrucker	1	399,00	399,00

Firma
Irma Computer GmbH&Co KG
Am Festungswall 45
65189 Wiesbaden

Rechnungsnummer:	R457	Rechnungsdatum:	04.03.1996
Kundennummer:	K195	Rechnungsbetrag:	359,00

Position	Artikelnummer	Bezeichnung	Anzahl	Einzelpreis	Gesamtpreis
1	A3117	CD-Rom	1	359,00	359,00

a) Stellen Sie fest, welche Objekte und Beziehungen sich daraus ableiten lassen und stellen Sie Ihre Ergebnisse in einer Tabelle der Form

Objekt | Beziehung | Eigenschaften

dar.

b) Formulieren Sie Geschäftsregeln für die Miniwelt.

c) Beschreiben Sie Objekte und Beziehungen durch ihre Eigenschaften und erstellen Sie entsprechend ein Modell der Miniwelt, indem Sie ein ER-Modell erstellen.

Aufgabe 2:

Alle im deutschen Sprachraum erscheinenden Bücher werden für die Deutsche Bibliothek nach einheitlichen Maßstäben gekennzeichnet. Die beiden folgenden Angaben stammen aus Büchern zu unserem Thema.

> Moos, Alfred:
> SQL-Datenbanken: der Weg vom Konzept zur Realisierung in dBase; eine schrittweise und praxisnahme Einführung/
> Alfred Moos; Gerhard Daues. - Braunschweig: Vieweg, 1991
> ISBN 3-528-05183-3
> NE: Daues, Gerhard (NE=Nebenautor)

> Heuer, Andreas:
> Objektorientierte Datenbanken: Konzepte, Modelle, Systeme/
> Andreas Heuer: - Bonn; München; Paris [u.a.]: Addison-Wesley, 1992
> ISBN 3-89319-315-4

Autoren können dabei als Einzelautor, Nebenautor oder als Herausgebern fungieren. Ein Buch kann beim gleichen Verlag in mehrfachen Ausgaben oder bei verschiedenen Verlagen erscheinen.

a) Erstellen Sie ein Modell der Miniwelt „Bibliothek“, indem Sie die Entities *Autor*, *Buch* und *Verlag* sowie die Beziehungen *hat_Urheberschaft_als* für die Autorenschaft und *ist_Ausgabe_von* für die Verlagsausgabe verwenden!

b) Formulieren Sie entsprechende Geschäftsregeln, z.B. hat ein Buch mindestens einen Autor, ein Autor kann verschiedene Bücher schreiben.
Was wäre an Ihrem Modell, welches auf den obigen Daten basiert, noch zu verbessern, wenn es um die eindeutige Identifizierung von Datensätzen geht?

Lösungen

Zur Modellierung von Ausschnitten der Wirklichkeit

Aufgabe 1:

a)
Aus den Beispielrechnungen lassen sich folgende Informationen über die Objekte *Kunde*, *Rechnung*, *Artikel* ableiten:

Kunde: Vorname, Name, Straße und Hausnummer, Plz, Ort, Kundennummer
Rechnung: Rechnungsnummer, Rechnungsdatum, Rechnungsbetrag
Artikel: Artikelnummer, Artikelbezeichnung, Artikelpreis

Die Informationen Positionsnummer, Anzahl und Postenpreis lassen sich keinem dieser Objekte direkt zuordnen, sie gehören zu der Beziehung **Rechnung_enthält_Artikel.** Als weitere Beziehung lässt sich finden **Kunde_erhält_Rechnung.**
Aus den erhaltenen Objekten und ihren Beziehungen lässt sich die folgende Tabelle herleiten:

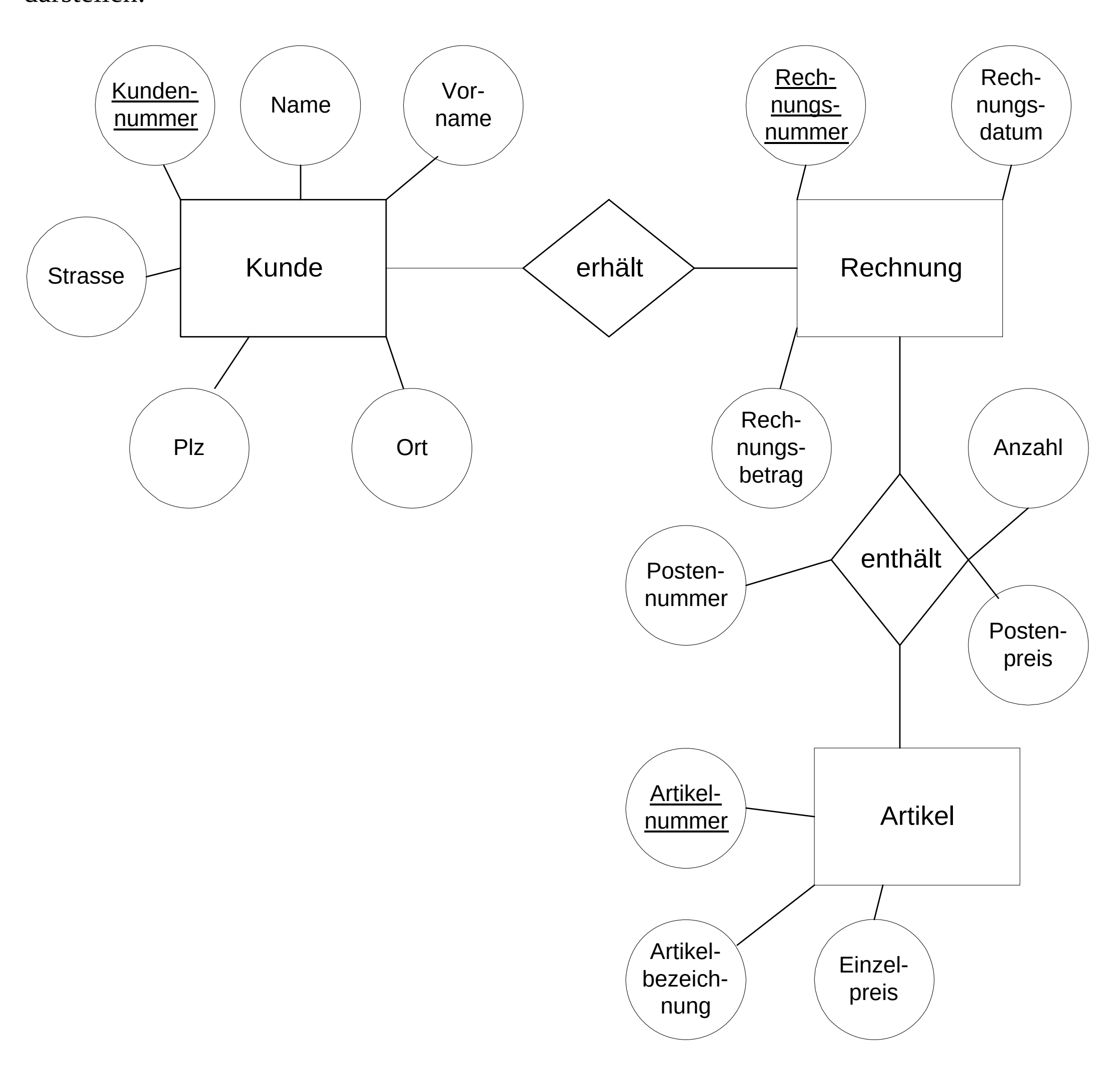
Objekt	Beziehung	Eigenschaften
Kunde		Kundennummer Name Vorname Strasse Plz Ort
	Kunde/Rechnung erhält	
Rechnung		Rechnungsnummer Rechnungsdatum Rechnungsbetrag
	Rechnung/Artikel enthält	Positionsnummer Anzahl Postenpreis
Artikel		Artikelnummer Artikelbezeichnung Einzelpreis

b)
Die Geschäftsregeln sind Aussagen über den Geschäftsgang und die Zusammenhänge der Elemente der Miniwelt. Es lassen sich folgende Aussagen machen:

- Eine Rechnung geht genau an einen Kunden, ohne Kunde existiert keine Rechnung
- Ein Kunde kann keine, eine oder mehrere Rechnungen erhalten.
- Eine Rechnung enthält mindestens eine Rechnungsposition. Ohne Rechnung kann eine Rechnungsposition nicht existieren.
- Eine Rechnungsposition betrifft genau einen Artikel und kann ohne diesen nicht existieren.

Ein Artikel kann in keiner, einer oder mehreren Rechnungspositionen erscheinen

c)
Das Entity-Relationship-Modell lässt sich anhand der Lösung zu a) sofort in grafischer Form darstellen:



Kunde – Rechnung: 1:n
Rechung – Artikel: m:n

Aufgabe 2:

a) Aus den beiden Buchangaben und den Hinweisen zur Urheberschaft und Herausgabe lassen sich folgende Entity-Typen und Beziehungs-Typen definieren:

Autor: Name, Vorname
Buch: Titel, Untertitel
hat_Urheberschaft_als: Autorenschaft (mit der Wertemenge *Hauptautor*, *Nebenautor*, *Herausgeber*)
Verlag: Verlagsbezeichnung, Verlagsorte (Mehrfachattribut)
ist_Ausgabe_von: Jahrgang, Auflage, Isbn

b) Als Geschäftsregeln lassen sich formulieren:

- Ein Buch besitzt mindestens einen Autor.
- Es existiert kein Autor ohne zugehöriges Buch.
- Ein Autor kann an einem oder mehreren Büchern beteiligt sein.
- Jedes Buch besitzt mindestens einen Verlag, der es herausgibt. Es gibt keinen Verlag ohne mindestens eine Buchausgabe.

c) Das ERM-Diagramm lässt sich sehr einfach zeichnen, da die Entity- und Beziehungs-Typen klar definiert sind und alle Eigenschaften erfasst wurden. Die eindeutigen Schlüssel sind noch nicht definiert. Hierzu eignen sich die erfassten Eigenschaften nicht, da sie nicht eindeutig sind. Es empfiehlt sich die Vergabe von eindeutigen Schlüsseln, z.B. Autorennummer, Buchnummer, Verlagsnummer

Aufgaben

Zur Komplexität von Beziehungen

Aufgabe 1:

Die folgende Tabelle zeigt jeweils zwei Entity-Typen und den zugehörigen Beziehungstyp. Geben Sie für jede Beziehung die Komplexität an.

(1)	Schüler	hat	Tutor
(2)	Schüler	bekommt heute	Zeugnis
(3)	Schüler	darf arbeiten an	Computer
(4)	Schüler	hat ausgeliehen	Buch
(5)	Schüler	besucht	Kurs
(6)	Schüler	ist befreundet mit	Schüler

Aufgabe 2:

Überlegen Sie, welche Komplexität die IS-A-Beziehung besitzt.

Aufgabe 3:

Gegeben sind jeweils zwei Entitytypen und ein Beziehungstyp. Geben Sie die jeweilige Komplexität an.

E-Typ 1	E-Typ 2	Beziehungs-Typ
Vater	Tochter	hat
Fuß	Zehe	gehört zu
Onkel	Neffe	hat
Schüler	Lehrer	hat Unterricht
Person	Personalausweis	besitzt
PROGRAM	FUNCTION	benötigt
Bruder	Schwester	hat
Ort	Ort	kürzeste Entfernung

Aufgabe 4:

Gegeben sind die Objekte
Klasse, Klasse, Klasse, Klassenleiter, Schüler, Lehrer
und die Beziehungen
hat, enthält, unterrichtet.

Stellen Sie für je 2 Objekte je eine Beziehung her und geben Sie die Kardinalität an.

Lösungen

Zur Komplexität von Beziehungen

Aufgabe 1:

Die folgende Tabelle zeigt jeweils zwei Entity-Typen und den zugehörigen Beziehungstyp. Geben Sie für jede Beziehung die Komplexität an.

(1)	Schüler	hat Tutor	n:1
(2)	Schüler	bekommt heute Zeugnis	1:1
(3)	Schüler	darf arbeiten an Computer	n:m
(4)	Schüler	hat ausgeliehen Buch	1:n
(5)	Schüler	besucht Kurs	n:m
(6)	Schüler	ist befreundet mit Schüler	n:m

Aufgabe 2:

Überlegen Sie, welche Komplexität die IS-A-Beziehung besitzt.

Die Antwort ist nicht ganz so einfach, wie man zuerst vermuten könnte. Im Beispiel zur Schulverwaltung ist ein Hausmeister auch genau ein Angestellter und ein Angestellter auch genau ein Mitglied der Schulgemeinde. Es handelt also hier um 1:1 - Beziehungen. Da wir jedoch mit IS-A - Beziehungen ganz allgemein Generalisierung und Spezialisierung erfassen wollen, sind auch 1:n - Beziehungen denkbar. Ein Bsp. wäre die Beziehung zwischen dem Ober-Entity-Typ *Autotyp* und dem Untertyp *Auto*. Zu *Autotyp* würden Attribute wie *Bezeichnung, PS-Zahl, Sitzplätze* usw. gehören, zu Auto z.B. *Kennzeichen, Zulassungsdatum, gefahrene km*. *Auto* erbt alle Attribute von *Autotyp*, aber es gibt natürlich mehrere Autos desselben Typs.

Aufgabe 3:

Gegeben sind jeweils zwei Entitytypen und ein Beziehungstyp. Geben Sie die jeweilige Komplexität an.

E-Typ 1	E-Typ 2	Beziehungs-Typ	
Vater	Tochter	hat	1:n
Fuß	Zehe	gehört zu	n:1
Onkel	Neffe	hat	n:m
Schüler	Lehrer	hat Unterricht	n:m
Person	Personalausweis	besitzt	1:1
PROGRAM	FUNCTION	benötigt	1:n
Bruder	Schwester	hat	n:m
Ort	Ort	kürzeste Entfernung	1:1

Aufgabe 4:

Gegeben sind die Objekte
Klasse, Klasse, Klasse, Klassenleiter, Schüler, Lehrer
und die Beziehungen
hat, enthält, unterrichtet.

Stellen Sie für je 2 Objekte je eine Beziehung her und geben Sie die Kardinalität an.

Klasse______1____ HAT _______________1____Klassenleiter
Klasse______1____ENTHÄLT___________n____Schüler
Lehrer______n____UNTERRICHTET____m___Klasse

Aufgaben

Zum Entity-Relationship-Modell

Aufgabe 1:

Der folgende Ausschnitt aus einem Kursverzeichnis dient zur Kurswahl der Schülerinnen und Schüler zur Jahrgangsstufe 13. Ergänzen Sie mit den darin befindlichen Kursdaten unser Modell zur Kursverwaltung und stellen Sie fest, welche Entities bzw. Entity-Eigenschaften hinzugefügt werden müssen!

Leiste G6: Mi 9./10. Std. und Fr 5. Std

5 D 13	Kreß	165	Literatur und Gesellschaft
5 D 16	Biedermann	164	Literatur und Gesellschaft
5 E 12	Meyer K.	127	The Individual and Society
5 K 10	Blum	7	Kunst und Gesellschaft

Leiste G8: Do 1.2. Std. und FR 6. Std

5 D 10	Biedermann	170	Literatur und Gesellschaft
5 D 17	Beste	164	Angst
5 E 11	Meyenburg	125	The Individual and Society
5 K 11	Repert	12	Kunst und Gesellschaft
5 Mu 10	Lamprecht	140	Musik und Gesellschaft

Aufgabe 2:

Geben sie die Komplexität der folgenden Situationen an. Erstellen Sie jeweils ein ER-Diagramm. Formulieren Sie sinnvolle Geschäftsbedingungen.

a) Eine Ladenkette möchte Informationen über ihre Filialen und über deren Zulieferer speichern. Jeder Zulieferer beliefert mehrere Filialen, und jede Filiale kauft von mehreren Zulieferern.

b) Ein Installateur möchte über seine Handwerker und über die Häuser, in denen sie gerade arbeiten, Informationen speichern.

Aufgabe 3:

Die Computerzubehörfirma Microtec GmbH möchte ihre Verwaltung auf EDV umstellen. Sie verkauft ein Sortiment von Artikeln, die sie von verschiedenen Herstellern bezieht. Außerdem hat sie einen bestimmten Kundenkreis, der bei ihr Bestellungen aufgibt. Eine Bestellung kann natürlich mehrere Artikel umfassen. Derselbe Artikel kann oft von mehreren Herstellern bezogen werden, und ein Hersteller liefert natürlich meist mehr als einen Artikel.
Erstellen Sie im Entity-Relationship-Modell ein sinnvolles Datenmodell für die Firma, das Datenredundanz vermeidet. Wählen Sie geeignete Entities mit notwendigen Attributen und geben Sie die zwischen den Entities bestehenden Beziehungen mit ihrem Komplexitätsgrad an.

Aufgabe 4:

Zugrunde gelegt werde das ER-Modell der Schulverwaltung, wie es im Text aufgestellt wurde. Geben Sie bei den nachfolgenden Änderungen an, welche Ebene des DBMS von der Änderung betroffen ist.

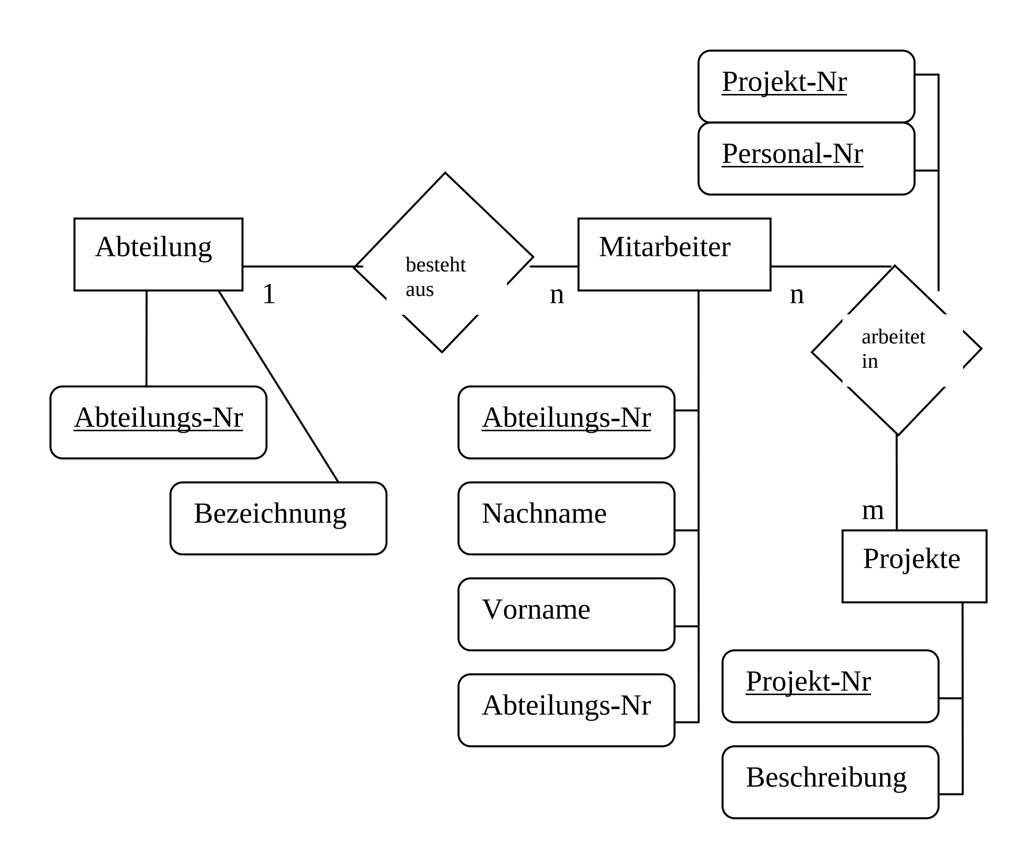
Der Lehrer Franz Schlauspruch kommt neu an die Schule.
Die Lehrer sollen bei der Noteneingabe nicht mehr die Noten des Schülers bei anderen Lehrern abfragen können.
Bei den Schülerdaten soll durch eine zusätzliche Indexdatei, in der die Schüler nach Wohnort sortiert sind, ein schnellere Suche nach Schülern eines Ortes ermöglicht werden, um die Busverbindungen besser koordinieren zu können.
Zur Erstellung von Alterstatistiken soll auch bei Lehrern das Geburtsdatum gespeichert werden.
Der Raum 556 soll als neuer Fachraum für Mathematik verwendet werden.
Auf Wunsch des Hausmeisters wird zusätzlich erfasst, welcher Lehrer einen Schlüssel für welchen Fachraum hat.
Neben dem/der Oberstufenleiter/in sollen auch die Tutoren die komplette Belegung ihrer Schüler einschließlich der bisher vergebenen Noten am Computer einsehen können.
Vom Sekretariat sollen die Schüler, die mehr als drei Grundkurse unter 5 Punkten einbringen müssen, per Serienbrief auf die Gefahr der Nichtzulassung zum Abitur hingewiesen werden.
Auf Antrag der SV dürfen Lehrer, die nicht Tutor des entsprechenden Schülers sind, die Fehlstundenzahl nicht mehr einsehen.
Daraufhin beschließt die Schulkonferenz, die Fehlstundenspeicherung ganz abzuschaffen.

Aufgabe 5:

Hugo Unbedarft besitzt eine große Spedition. Er will seine Auftragsverwaltung auf EDV umstellen und macht sich dazu einen genauen Plan. Seine Aufträge sind immer so, dass sie nur zu einem Ziel führen, es kann allerdings möglich sein, dass mehrere LKWs für einen Auftrag nötig sind. Nicht jeder LKW-Typ ist dazu geeignet, alle Ziele zu erreichen (z.B. zu niedrige Brücken), und nicht jeder Fahrer kann jeden LKW-Typ fahren. Hugo will folgende Daten speichern: AuftragsNr. und LKW-Nr., Ziel, Zielentfernung, Auftragsdatum, LKW-Typ, max. Zuladung eines LKW-Typs, TÜV-Datum, Fahrer-Nr, Fahrer-Name.
Erstellen Sie ein ER-Modell für die Spedition.

Aufgabe 6:

Gegeben ist folgendes ERM:



a) Leiten Sie daraus die kompakte ERM-Notation ab: Tabellenname(Primärschlüssel, Attribute)

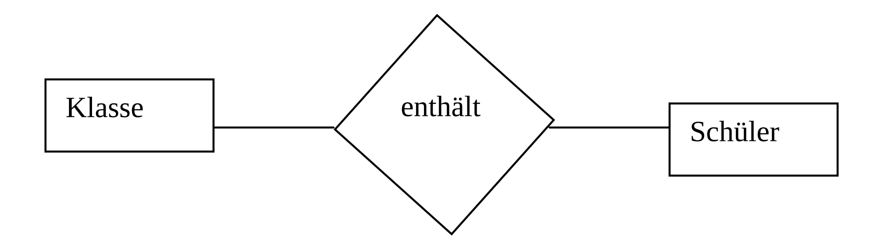
b) Geben Sie die Schritte an, wie aus einem ERM ein relationales Modell wird.

Aufgabe 7:

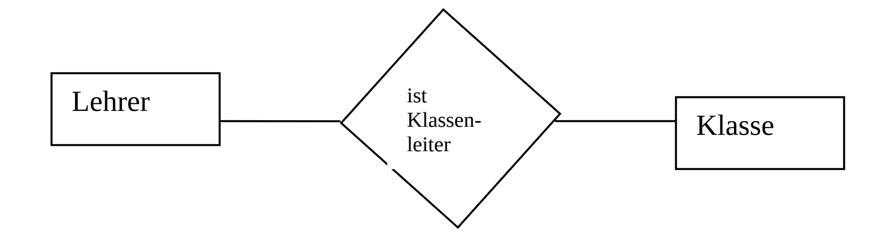
Wir bauen nun sukzessive ein vereinfachtes Lehrer-Schüler Unterrichtsmodell auf.

a) Bestimmen Sie die beteiligten Objekte.
b) Ermitteln Sie für alle Schritte die Kardinalitäten.
c) Leiten Sie nun einmal versuchsweise das ERM her. Details folgen im nächsten Kapitel. Versehen Sie dabei die Objekte mit sinnvollen Attributen.

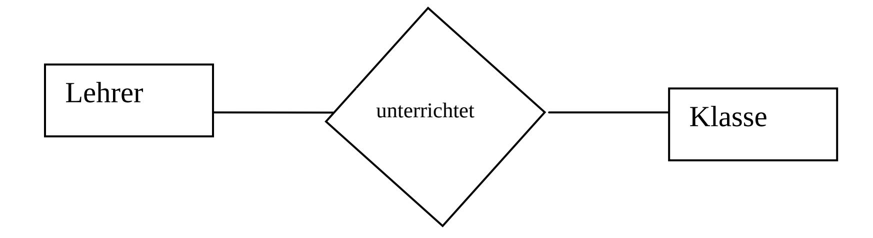
Schritt 1:



Schritt 2:



Schritt 3:



Lösungen

Zum Entity-Relationship-Modell

Aufgabe 1:

Der folgende Ausschnitt aus einem Kursverzeichnis dient zur Kurswahl der Schülerinnen und Schüler zur Jahrgangsstufe 13. Ergänzen Sie mit den darin befindlichen Kursdaten unser Modell zur Kursverwaltung und stellen Sie fest, welche Entities bzw. Entity-Eigenschaften hinzugefügt werden müssen!

Leiste G6: Mi 9./10. Std. und Fr 5. Std

5 D 13	Kreß	165	Literatur und Gesellschaft
5 D 16	Biedermann	164	Literatur und Gesellschaft
5 E 12	Meyer K.	127	The Individual and Society
5 K 10	Blum	7	Kunst und Gesellschaft

Leiste G8: Do 1.2. Std. und FR 6. Std

5 D 10	Biedermann	170	Literatur und Gesellschaft
5 D 17	Beste	164	Angst
5 E 11	Meyenburg	125	The Individual and Society
5 K 11	Repert	12	Kunst und Gesellschaft
5 Mu 10	Lamprecht	140	Musik und Gesellschaft

Die meisten Angaben sind im ERM-Diagramm Schule bereits enthalten, d.h. unser Modell bildet die Realität bereits weitgehend richtig ab. Die Eigenschaften des Entity-Typs *Kurs* sind mit Kurs-Nr., Jahrgangsstufe, Fach und Thema vollständig, die Angaben über den Unterrichtenden erhält man über den Beziehungs-Typ *Lehrer_hält_Kurs*. Über den Beziehungs-Typ *Kurs_in_Raum* ist die Eigenschaft Raum-Nr. zugänglich. Lediglich das Mehrfachattribut Zeiten bildet die Realität nicht korrekt ab, weil Leistungs- und Grundkurs-Leisten nicht erfaßt sind. Da zu jeder Leiste mehrere Zeiten gehören, ist die Einfügung eines neuen Entity-Typs *Leiste* mit den Attributen Leistenbezeichnung und Zeiten erforderlich. Den Zusammenhang stellt der neue Beziehungs-Typ *Kurs_belegt_Leiste* her.

Aufgabe 2:

Geben sie die Komplexität der folgenden Situationen an. Erstellen Sie jeweils ein ER-Diagramm. Formulieren Sie sinnvolle Geschäftsbedingungen.

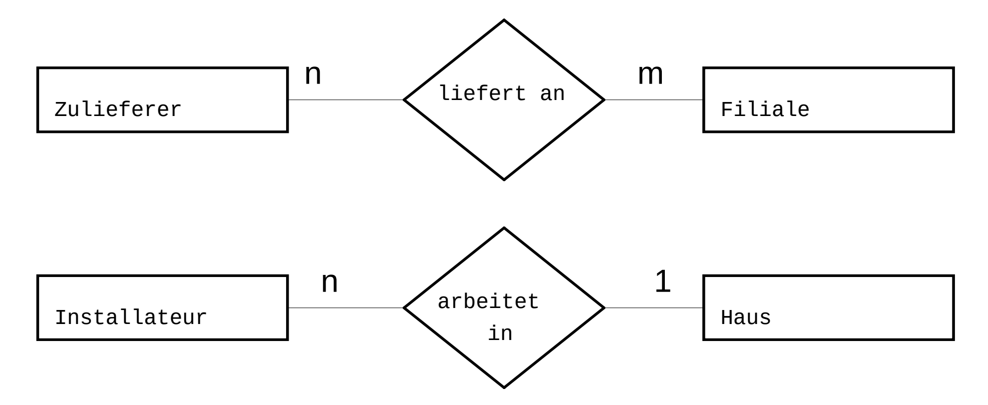
a) Eine Ladenkette möchte Informationen über ihre Filialen und über deren Zulieferer speichern. Jeder Zulieferer beliefert mehrere Filialen, und jede Filiale kauft von mehreren Zulieferern.

Es handelt sich um eine m:n Beziehung.
Sinnvolle Geschäftsregeln:
- Alle Filialen werden von mindestens einem Zulieferer beliefert.
- Es gibt auch Zulieferer, die zur Zeit keine Filiale beliefern, sie bleiben aber trotzdem für spätere Fälle in der Datei.

b) Ein Installateur möchte über seine Handwerker und über die Häuser, in denen sie gerade arbeiten, Informationen speichern.

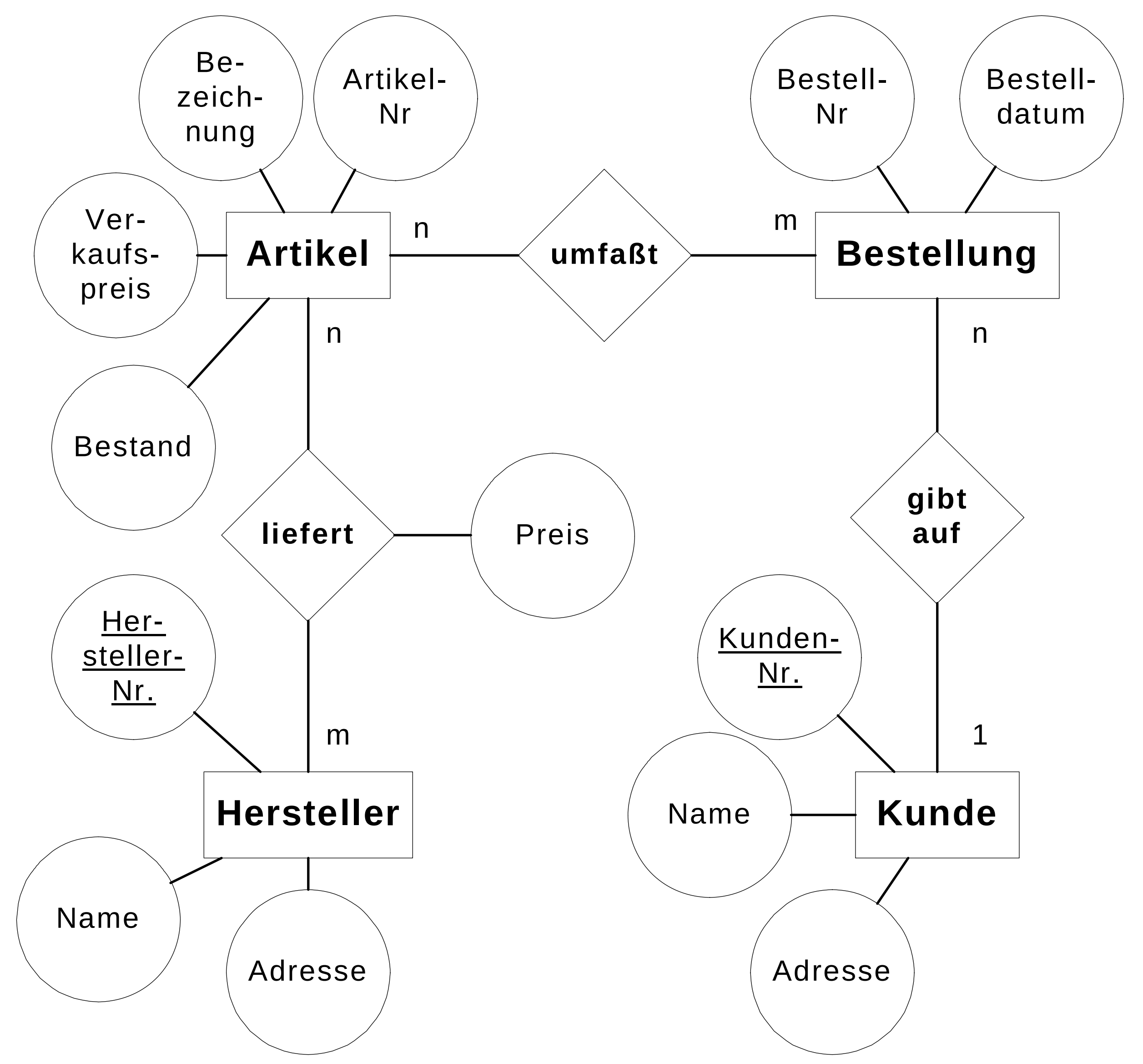
Wir haben hier angenommen, dass jeder Installateur zur Zeit nur in einem Haus arbeitet, dass aber in jedem Haus mehrere Installateure arbeiten. Wegen Krankheitsfällen oder anderer Aufgaben sollte es zulässig sein, dass ein Installateur in keinem Haus arbeitet, wegen Lieferschwierigkeiten oder witte-

rungsbedingt sollte es zulässig sein, dass in einem Haus nicht gearbeitet wird. Es handelt sich um eine n : 1 Beziehung.



Aufgabe 3:

Eine mögliche Lösung ist als ERM grafisch dargestellt:



Adresse wird bei der konkreten Umsetzung noch unterteilt in Strasse/Nr/PLZ/Ort

Aufgabe 4:

(1) Kein Modell ist betroffen. Es werden lediglich zusätzliche Daten eingegeben.
(2) Externe Sicht für die Benutzer „Lehrer“ wird geändert.
(3) Die interne Organisation der Daten im Rechner wird geändert, also ist das interne Modell betroffen.
(4) Das konzeptuelle Modell muss geändert werden. Der Entity-Typ Lehrer erhält ein zusätzliches Attribut.
(5) Kein Modell ist betroffen, es werden lediglich Dateneinträge geändert.
(6) Das konzeptuelle Modell wird geändert. Zwischen Lehrer und Fachraum wird eine neue Beziehung *hat Schlüssel für* aufgebaut.
(7) Die externe Sicht für die Benutzer „Tutoren“ wird geändert.
(8) Etwas unklares Beispiel. Es wäre denkbar, dass die externe Sicht für den Benutzer „Textverarbeitungsprogramm im Sekretariat“ geändert werden muss.
(9) Externe Sicht für die Benutzer „Lehrer“ wird geändert.
(10) Das konzeptuelle Modell ist betroffen. Das Attribut „Fehlstunden“ der Beziehung *besucht* wird gelöscht.

Aufgabe 5:

Es lassen sich folgende Objekte und Beziehungen feststellen:
Auftrag wird durchgeführt von LKW n:m-Beziehung
Begründung: Ein Auftrag wird von einem oder mehreren LKWs durchgeführt; ein LKW kann an keinem, einem oder mehreren Aufträgen beteiligt sein.

LKW wid gefahren von Fahrer :m-Beziehung
Begründung: Ein LKW wird von keinem, einem oder mehreren Fahrern gefahren; ein Fahrer kann verschiedene LKWs fahren (natürlich nicht gleichzeitig).

Die Attribute für die Objekte ergeben sich direkt aus dem Aufgabentext.

Aufgabe 6:

a) Leiten Sie daraus die kompakte ERM-Notation ab: Tabellenname(Primärschlüssel, Attribute)

aus der 1:n Beziehung 1:1 übernommen
ABTEILUNG(<u>Abteilungs-Nr</u>, Bezeichnung)
MITARBEITER(<u>Personal-Nr</u>, Nachname, Vorname, Abteilungs-Nr)
Zwischen MITARBEITER und PROJEKT besteht eine n:m Beziehung, die Aufgelöst werden muss im relationalen Modell. Übernehme in eine Beziehungs-tabelle die PS der beteiligten Entities. Also
PROJEKTAUSWERTUNG(<u>Projekt-Nr</u>, <u>Personal-Nr</u>)
PROJEKT(Projekt-Nr, Beschreibung)

b) Geben Sie die Schritte an, wie aus einem ERM ein relationales Modell wird.

1. Aus jeder Entitätsmenge kann eine Tabelle direkt abgeleitet werden.
2. Jede 1:n-Beziehung kann sich direkt in entsprechende Tabellen nieder.
3. Jede n:m-Beziehung muss aufgelöst werden in zwei 1:n-Beziehungen. Dabei enstehen dann aus zwei Tabellen drei, denn es wird eine Beziehungstabelle als n-Seite erstellt, die die PS der beteiligten Tabellen enthält und die Verknüpfung herstellt.
4. Die Beziehungen werden über die entsprechenden Verbindungen im relationalen Modell hergestellt.
5. Datenintegrität wird per referentieller Integrität gesichert.

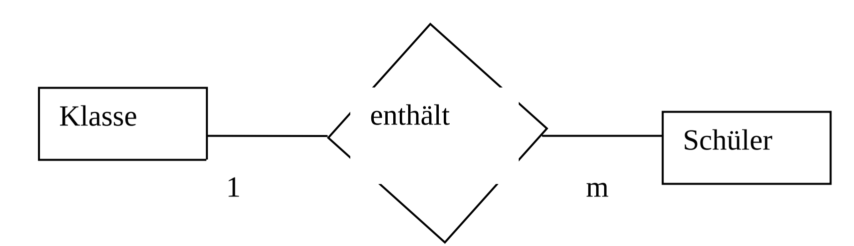
Aufgabe 7:

Wir bauen nun sukzessive ein vereinfachtes Lehrer-Schüler Unterrichtsmodell auf.

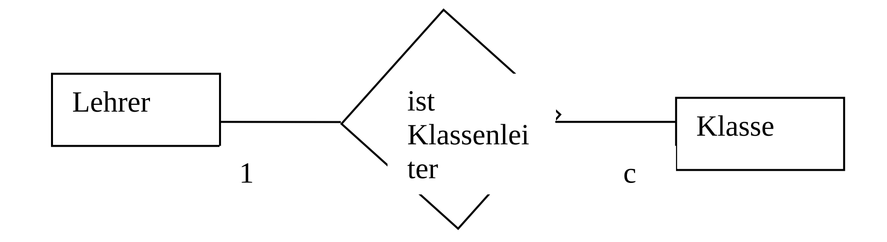
a) Bestimmen Sie die beteiligten Objekte.
b) Ermitteln Sie für alle Schritte die Kardinalitäten.
c) Leiten Sie nun einmal versuchsweise das ERM her. Details folgen im nächsten Kapitel. Versehen Sie dabei die Objekte mit sinnvollen Attributen.

a) Klasse, Schüler, Lehrer

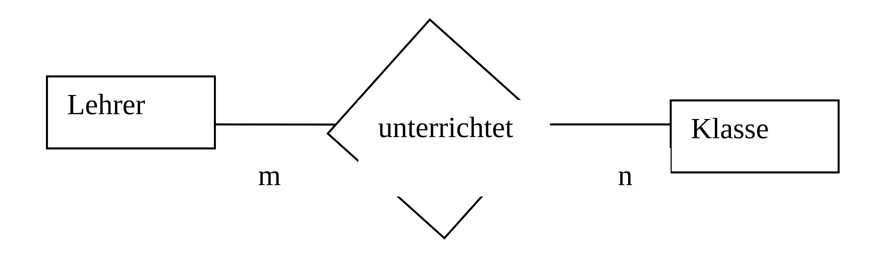
Schritt 1:



Schritt 2:

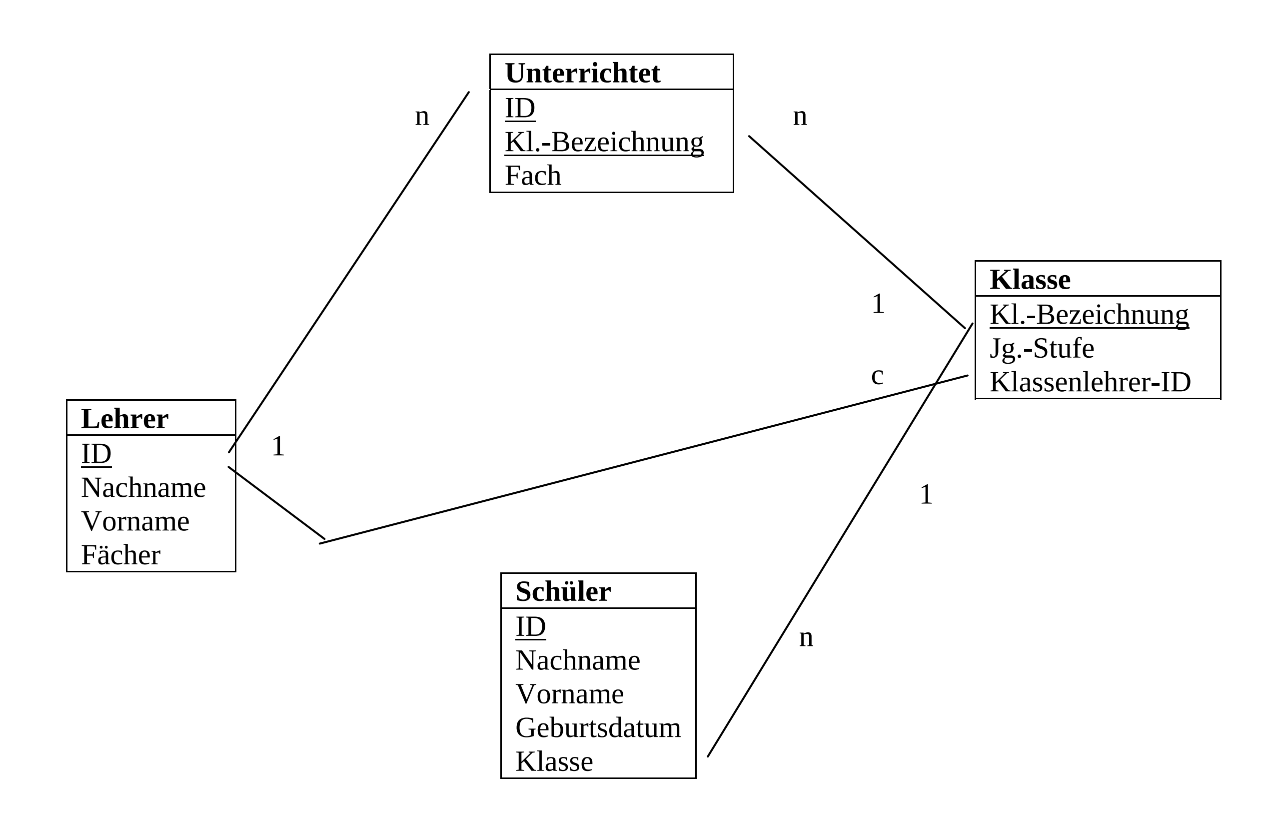


Schritt 3:



Auf dieser Basis wird nun das ERM aufgebaut.

Schritt 1: 1:m
2 Tabellen, keine extra Beziehungsrelation
Schritt 2: 1:c
2 Tabellen, keine extra Beziehungsrelation
Schritt 3: m:n
Auftrennen in zwei 1:n-Beziehungen, also insgesamt 3 Tabellen; die Beziehungsrelation wird in eine eigene Tabelle abgebildet und enthält die Primärschlüssel der beteiligten Entities als zusammengesetzter Primärschlüssel.


Unterrichtet
ID
Kl.-Bezeichnung
Fach
n
n
Klasse
Kl.-Bezeichnung
Jg.-Stufe
Klassenlehrer-ID
1
c
Lehrer
ID
Nachname
Vorname
Fächer
1
1
Schüler
ID
Nachname
Vorname
Geburtsdatum
Klasse
n

3 Das relationale Datenbankmodell

Bisher liegen die Daten unserer Miniwelt in einem abstrakten Modell vor, das die Realität durch Entitytypen und ihre Beziehungen zueinander widerspiegelt. Das ER-Modell ist - wie wir gesehen haben - in der Phase des konzeptionellen Entwurfs einer Datenbank hervorragend geeignet, ein Datenbanksystem ohne Abhängigkeit von der späteren Implementierung zu gestalten. Im Anschluss an den konzeptionellen Entwurf folgt beim Datenbankdesign der *logische Entwurf*. Hierbei wird ein spezielles Datenbankmodell ausgewählt und das Ergebnis des konzeptionellen Entwurfs auf dieses Datenbankmodell abgebildet. Meist ist noch eine Optimierung anhand unterschiedlicher Qualitätskriterien erforderlich.

Die klassischen DBMS-Systeme, die als Software auf Computern zur Verfügung stehen, beruhen alle auf Datenbankmodellen, deren Gestaltungsmöglichkeiten gegenüber dem ER-Modell eingeschränkt sind. Man unterscheidet zwischen dem hierarchischen, dem objektorientierten, dem Netzwerk- und dem relationalen Modell. Das mit Abstand wichtigste ist das *relationale Modell*, deshalb soll es hier als einziges vorgestellt werden. Es geht zurück auf Arbeiten von E. F. Codd Anfang der 70er Jahre. Auch *Access* verwendet dieses Datenmodell (ebenso wie *dBase, Paradox, Oracle* usw.). Das relationale Modell beruht auf einfachen mathematischen Grundlagen.

3.1 Konzepte des relationalen Datenbankmodells

Das relationale Datenbankmodell basiert auf dem mathematischen Begriff der *Relation.* Eine Relation R ist eine Teilmenge des kartesischen Produkts einer Liste von Wertbereichen $W_1, W_2, ..., W_n$:

$$R \subseteq W_1 \times W_2 \times ... \times W_n$$

Elemente einer Relation R sind die n-Tupel der Art $(v_1, v_2, ..., v_n)$ mit $v_i \in W_i$. Die Größe bezeichnet man als Grad der Relation R.

Im Beispiel der Oberstufenverwaltung kommen Kurse als Objekte vor. Kurse werden durch die drei Attribute *Kurs-Nr, Thema* und *Jahrgangsstufe* beschrieben. Die Kursnummer ist eine natürliche Zahl und das Thema eine Zeichenkette (String). Die Wertemengen sind demnach:

$$W_1 = N$$

$$W_2 = \{x \mid \text{ist String}\}$$

$$W_3 = \{11/I, 11/II, 12/I, 12/II, 13/I, 13/II\}$$

Exemplarisch sind hier vier Kurse K_1 bis K_4 in Form von Tripeln (3-Tupel) angegeben:

K_1 = (13, Analysis 3, 13/II)

K_2 = (2, Short Stories, 12/I)

K_3 = (38, Datenbanken, 12/II)

K_4 = (19, Antihelden, 11/II)

Sie bilden die vierelementige Relation

KURS = {(13,Analysis 3,13/II), (2,Short Stories,12/I), (38,Datenbanken,12/II), (19, Antihelden, 11/II}

Die Mengenschreibweise ist unübersichtlich. Daher ist die dazu äquivalente Tabellenschreibweise besser geeignet und üblich:

KURS

Kurs-Nr	Thema	Jahrgangsstufe
13	Analysis 3	13/II
2	Short Stories	12/I
38	Datenbanken	12/II
19	Antihelden	11/II

In der ersten Zeile der Tabelle werden die jeweiligen Attribute als Spaltenköpfe notiert. Sie gehören somit nicht zur Relation. Vielmehr beschreiben sie das *Relationenschema*, also den hier beschriebenen Entitytyp. Jede weitere Zeile der Tabelle beschreibt ein Tupel der Relation, entspricht also einem Entity. Der allgemeine Aufbau einer Tabelle im Relationenmodell ist demnach gegeben durch:

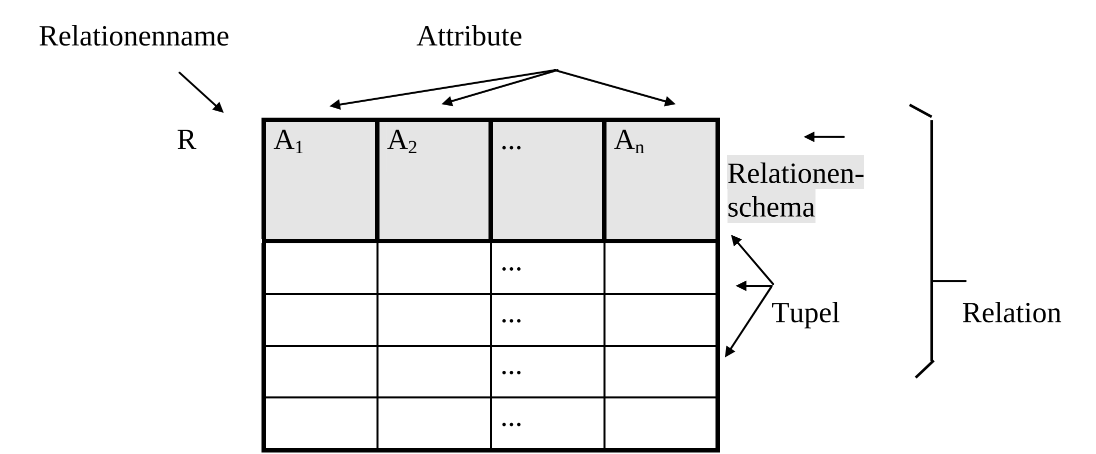


Tabelle 1: Relationenschema

Den neu eingeführten Begriffen stellen wir in folgender Tabelle eine erläuternde Beschreibung und eine Übersetzung in das ER-Modell gegenüber.

Relationales Modell	Beschreibung	ER-Modell	Darstellung
Relationenname	Name der Tabelle	Name des Entitytyps	Name im Rechteck
Attribut	Spalte einer Tabelle	Attribut	Oval
Relationenschema	Menge von Attributen	Entitytyp	Rechteck samt Ovalen
Tupel	Zeile einer Tabelle	Entity	-
Relation	Menge von Zeilen	Entities (Entitymenge)	-

Der Übersetzungstabelle vom relationalen Modell in das ER-Modell entnehmen wir, dass das Relationenschema dem Entitytyp des ER-Modells entspricht. Um über Relationenschemata sprechen zu können, benutzt man die Schreibweise:

Allgemein: R($\underline{A_1}$, A_2, ..., A_3)

Beispiel: Kurs(Kurs-Nr, Thema, Jahrgangsstufe)

Schlüsselattribute zeichnet man **durch Unterstreichung** aus.

Bei der Darstellung einer Relation als Tabelle muss man immer beachten, dass die Tabelle lediglich eine übersichtliche Darstellung einer Menge ist. Dies bedeutet unter anderem, dass in einer Relationentabelle nie zwei gleiche Zeilen vorkommen dürfen und die Reihenfolge der Zeilen keine Bedeutung hat.

3.2 Operatoren des Relationenmodells

Mit dem Tabellenkonzept sind wir in der Lage, Entitytypen in das Relationenmodell abzubilden. Der Abbildung von Beziehungen im ER-Modell werden wir uns später zuwenden. Um Missverständnissen vorzubeugen, sei betont, dass die *Beziehungen* im ER-Modell nicht dem entsprechen, was wir hier als *Relationen* bezeichnet haben!

Im Folgenden beschäftigen wir uns mit den Operatoren des Relationenmodells, mit denen neben den rein statischen Eigenschaften des ER-Modells auch dynamische Eigenschaften modelliert werden können. Die Operatoren können auf Relationen angewendet werden und erzeugen dabei neue Relationen. Wie wir aus der Mathematik wissen, lassen sich Operatoren und Operanden (Relationen) zu komplizierten Ausdrücken verknüpfen, mit denen die Berechnung neuer Relationen beschrieben werden kann. Man spricht daher auch von der *Relationenalgebra*. Sie ist eine präzise Sprache zur Formulierung von Anfragen, die noch unabhängig ist von der Sprache eines bestimmten Datenbankmanagementsystems. Man kann hier also Dinge noch anfragespracheunabhängig formulieren, egal in welchem DBMS man später implementieren will. Damit *leistet die Relationenalgebra die Verknüpfung unterschiedlicher Tabellen* und damit das logische wieder zusammen fügen dessen, was aus Modellierungsgründen getrennt wurde.

Bedeutsam sind primär die folgenden Operatoren:

Operator	Schreibweise	Bedeutung
Durchschnitt	$R \cap S$	Schnittmenge
Vereinigung	$R \cup S$	Vereinigungsmenge
Differenz	$R \setminus S$	Differenz von Mengen
Produkt	$R \times S$	kartesisches Produkt zweier Mengen
Selektion	$\sigma_{Formel}(R)$	Auswahl von Tupel gemäß Formel, Streichung von Zeilen
Projektion	$\pi_{Attribute}(R)$	Auswahl von Attributen, Streichung von Spalten
Join (natural)	$R \rhd\lhd S$	Verknüpfung zweier Relationen zu einer neuen mit den Attributen beider Tabellen über gemeinsames Attribut
Join (Theta)	$R \underset{C}{\rhd\lhd} S$	Verknüpfung zweier Relationen zu einer neuen mit formulierter Bedingung C

Dabei wird zwischen **Sets** und **Bags** unterschieden.

Sind Mehrfachvorkommen eines Tupels (Zeile) erlaubt, so spricht man von Bags (*Eselsbrücke*: Die Ergebnistupel einfach ohne Nachbearbeitung in die ,,Tasche (bag)" werfen). Werden Duplikate entfernt handelt es sich um Sets. Demnach werden temporär (zeitweilig) erst Bags gebildet, aus denen dann mehrfach vorkommende Elemente entfernt werden, um so als finales Ergebnis ein Set zu erhalten.

Wir gehen im Folgenden von Sets aus.

Zur Erklärung und Verdeutlichung der Wirkungsweise der Operatoren betrachten wir Beispiele mit folgenden Relationen:

Kurs1

Kurs-Nr	Thema	Jahrgangsstufe
13	Analysis	12/I
25	Short Stories	12/I
3	Datenbanken	11/I

Kurs2

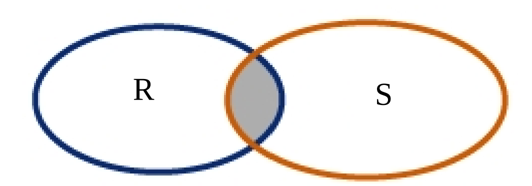
Kurs-Nr	Thema	Jahrgangsstufe
11	Mechanik I	11/I
12	Mechanik I	11/I
25	Short Stories	12/I
3	Datenbanken	11/I

KursLR

Kurs-Nr	Lehrer	Raum
11	Müller I	123
12	Schulze	124
27	Bauer	14
15	Maier	14
17	Maier	17
3	Zange	211

Durchschnitt

Der Durchschnitt R ∩ S zweier Relationen R und S ist die Menge aller Tupel, die sowohl in R als auch in S enthalten sind.

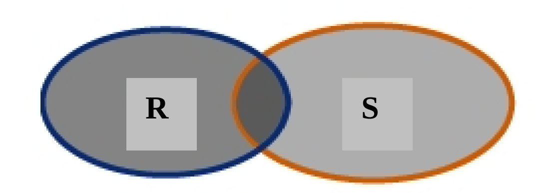


Kurs1 ∩ Kurs2

Kurs-Nr	Thema	Jahrgangsstufe
25	Short Stories	12/I
3	Datenbanken	11/I

Vereinigung

Die Vereinigung R ∪ S zweier Relationen R und S ist die Menge aller Tupel, die in R oder S oder in beiden Relationen enthalten sind.

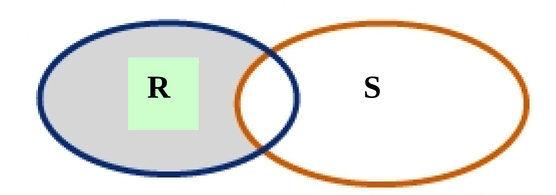


Kurs1 ∪ Kurs2

Kurs-Nr	Thema	Jahrgangsstufe
13	Analysis	12/I
25	Short Stories	12/I
3	Datenbanken	11/I
11	Mechanik I	11/I
12	Mechanik I	11/I

Differenz

Die Differenz R \ S zweier Relationen R und S ist die Menge aller Tupel, die in R aber nicht gleichzeitig in S enthalten sind.



Kurs1 \ Kurs2

Kurs-Nr	Thema	Jahrgangsstufe
13	Analysis	12/I

Kartesisches Produkt - Verbinden zweier Tabellen

Seien R und S Relationen mit Grad n_1 und n_2. Das kartesische Produkt R × S ist die Menge aller (n_1 * n_2)-Tupel, deren erste n_1 Komponenten ein Tupel in R und deren letzte n_2 Komponenten ein Tupel aus S darstellen. Also wird 'jedes Element mit jedem' gepaart.

Anm.: Wenn Spalten gleichen Namens in den Tabellen auftauchen, dann man sie auch über *Tabellenname.Spaltenname* unterscheiden, z.B. Kurs2.Kurs-Nr.

Kurs2 × KursLR

Kurs -Nr	Thema	Jahr-gangsstufe	Kurs-Nr	Lehrer	Raum
11	Mechanik I	11/I	11	Müller 1	123
11	Mechanik I	11/I	12	Schulze	124
11	Mechanik I	11/I	27	Bauer	14
11	Mechanik I	11/I	15	Maier	14
11	Mechanik I	11/I	17	Maier	17
11	Mechanik I	11/I	3	Zange	211
12	Mechanik I	11/I	11	Müller 1	123
12	Mechanik I	11/I	12	Schulze	124

12	Mechanik I	11/I	27	Bauer	14
12	Mechanik I	11/I	15	Maier	14
12	Mechanik I	11/I	17	Maier	17
12	Mechanik I	11/I	3	Zange	211
25	Short Stories	12/I	11	Müller 1	123
25	Short Stories	12/I	12	Schulze	124
25	Short Stories	12/I	27	Bauer	14
25	Short Stories	12/I	15	Maier	14
25	Short Stories	12/I	17	Maier	17
25	Short Stories	12/I	3	Zange	211
3	Datenbanken	11/I	11	Müller 1	123
3	Datenbanken	11/I	12	Schulze	124
3	Datenbanken	11/I	27	Bauer	14
3	Datenbanken	11/I	15	Maier	14
3	Datenbanken	11/I	17	Maier	17
3	Datenbanken	11/I	3	Zange	211

Selektion - Auswahl von Zeilen

Sei F eine Formel. Diese Formel kann Konstanten und Attribute als Operanden, sowie Vergleichsoperatoren ($<$, $\leq$, $=$, $\neq$, $\geq$, $>$) und logische Operatoren (and, or, not) enthalten. Dann ist die Selektion σ_F (R) die Menge aller Tupel in R, die die Formel F erfüllen.

$\sigma_{\text{Kurs-Nr} > 15}$ (KursLR)

Kurs-Nr	Lehrer	Raum
27	Bauer	14
17	Maier	17

Projektion - Auswahl von Spalten

Sei R eine Relation vom Grad n. Dann ist $\pi_{i_1}, \pi_{i_2},..., \pi_{i_m}(R)$ die Abbildung von R auf die Komponenten $i_1, i_2, ..., i_m$. Die entstehende Relation ist m-gradig.

$\pi_{\text{Thema,Jahrgangsstufe}}(\text{Kurs1} \cup \text{Kurs2})$

Kurs-Nr	Thema	Jahrgangsstufe
13	Analysis	12/I
25	Short Stories	12/I
3	Datenbanken	11/I
11	Mechanik 1	11/I
12	Mechanik 1	11/I

Duplikate müssen aus der projizierten Tabelle entfernt werden, denn sie muss eine Menge darstellen:

$\pi_{Thema,Jahrgangsstufe}(Kurs1 \cup Kurs2)$

Thema	Jahrgangsstufe
Analysis	12/I
Short Stories	12/I
Datenbanken	11/I
Mechanik 1	11/I

Join (natürlicher Verbund) - Tabellen verbinden

Der Join R⊳⊲S der Tabellen R und S wird für Tabellen mit gleichen Attributen wie folgt berechnet:

Man bildet das kartesische Produkt $R \times S$

Für jedes Attribut, das sowohl in R als auch in S vorkommt, selektiert man die Tupel, für die die Werte der gleichnamigen Attribute übereinstimmen.

Eine der gleichen Spalten wird wegprojiziert.

Als Merkregel formuliert:

Der Join verbindet zwei Tabellen über gleichnamige Spalten bei gleichen Attributwerten.

Der natürliche Verbund ist äußerst wichtig, um Relationen, die aus entwurfstheoretischen Gründen zerlegt wurden, während der Abfrage wieder zu kombinieren. Die Zerlegung findet in der Regel über Schlüsselattribute statt. Dementsprechend findet der Join in aller Regel über ein gemeinsames Schlüsselattribut in den beiden zu verknüpfenden Tabellen statt.

In unserem Beispiel werden durch den Join die Kurs-Informationen mit den Lehrer- und Raum-Informationen verknüpft. So entsteht aus der Kurs-Tabelle und der Lehrer-Raum-Tabelle eine neue Kurs-Lehrer-Raum-Tabelle:

Kurs2 ⊳⊲ KursLR

Kurs-Nr	Thema	Jahrgangsstufe	Lehrer	Raum
11	Mechanik 1	11/I	Müller 1	123
12	Mechanik 1	11/I	Schulze	124
3	Datenbanken	11/I	Zange	211

Zur Berechnung des Join:

Als erstes wird das kartesische Produkt der Ausgangstabellen Kurs2 und KursLR gebildet, die entstehende Tabelle besteht aus 4*6 = 24 Tupeln. Diejenigen Tupel, die bezüglich des gemeinsamen Attributs *Kurs-Nr* gleiche Werte aufweisen, werden selektiert. Zuletzt wird die doppelte Spalte *Kurs-Nr* ausprojiziert und das obige Ergebnis hergestellt. Die folgende Abbildung zeigt diese Vorgehensweise nochmals.

Kurs2 $\rhd\lhd$ KursLR

Kurs-Nr	Thema	Jahrgangsstufe	Kurs-Nr	Lehrer	Raum
11	Mechanik I	11/I	11	Müller 1	123
11	Mechanik I	11/I	12	Schulze	124
11	Mechanik I	11/I	27	Bauer	14
11	Mechanik I	11/I	15	Maier	14
11	Mechanik I	11/I	17	Maier	17
11	Mechanik I	11/I	3	Zange	211
12	Mechanik I	11/I	11	Müller 1	123
12	Mechanik I	11/I	12	Schulze	124
12	Mechanik I	11/I	27	Bauer	14
12	Mechanik I	11/I	15	Maier	14
12	Mechanik I	11/I	17	Maier	17
12	Mechanik I	11/I	3	Zange	211
25	Short Stories	12/I	11	Müller 1	123
25	Short Stories	12/I	12	Schulze	124
25	Short Stories	12/I	27	Bauer	14
25	Short Stories	12/I	15	Maier	14
25	Short Stories	12/I	17	Maier	17
25	Short Stories	12/I	3	Zange	211
3	Datenbanken	11/I	11	Müller 1	123
3	Datenbanken	11/I	12	Schulze	124
3	Datenbanken	11/I	27	Bauer	14
3	Datenbanken	11/I	15	Maier	14
3	Datenbanken	11/I	17	Maier	17
3	Datenbanken	11/I	3	Zange	211

Join (Theta-Verbund) - Tabellen verbinden

Der natürliche Verbund vereint Tabellen über gemeinsame Attribute. Möchte man aber die Bedingungen selbst vorgeben, so führt dies zum Theata-Join, formuliert durch $R \underset{C}{\rhd\lhd} S$.

Dabei ist C die Bedingung (condition), die den Stellenwert der gemeinsamen Attribute beim natürlichen Verbund einnimmt.

3.3 Anwendung relationaler Operatoren

Zur Darstellung eines komplexeren Beispiels zur Relationenalgebra gehen wir von folgenden Tabellen einer Oberstufenverwaltung aus:

Schüler

Schüler-Nr	Name	Vorname	Tutor	Geschlecht
123	Alberti	Hans	Müller	m
034	Glücklich	Gesine	Abel	w
321	Müser	Angelika	Abel	w
111	Weber	Wolfgang	Zange	m

Kurs

Kurs-Nr	Typ	Fach	Thema	Jahrgangsstufe
13	GK	Mathematik	Analysis 2	12/I
11	GK	Physik	Mechanik 1	11/I
03	GK	Informatik	Datenbanken	12/II
25	LK	Englisch	Short Stories	12/I
89	GK	Informatik	Compilerbau	13/II

Besucht

Schüler-Nr	Kurs-Nr	Fehlstunden	Punkte
123	03	00	12
123	25	03	07
321	89	00	14
111	03	21	03

Es soll die Frage beantwortet werden, welche Mädchen Informatikkurse besuchen und welche Punktzahlen sie dabei erreicht haben.

1. Bestimmung der Informatikkurse

Informatikkurse = $\pi_{\text{Kurs-Nr}}(\sigma_{\text{Fach = Informatik}}(\text{Kurs}))$

Kurs-Nr
03
89

2. Join mit der *Besucht*-Tabelle über das gemeinsame Schlüsselattribut *Kurs-Nr* liefert die Informatikschüler

Informatikschüler = Informatikkurse $\bowtie$ Besucht

Kurs-Nr	Fach	Schüler-Nr	Fehlstunden	Punkte
03	Informatik	123	00	12
03	Informatik	111	21	03
89	Informatik	321	00	14

3. Projektion auf die benötigten Attribute *Schüler-Nr* und *Punkte*:

InformatikschülerPunkte = $\pi_{\text{Schüler-Nr, Punkte}}$ (Informatikschüler)

Schüler-Nr	Punkte
123	12
111	03
321	14

4. Join mit *Schüler*-Tabelle über das gemeinsame Schlüsselattribut *Schüler-Nr*.

InformatikschülerPunkteName = InformatikschülerPunkt $\bowtie$ Schüler

Schüler-Nr	Punkte	Name	Vorname	Tutor	Geschlecht
123	12	Alberti	Hans	Müller	m
111	03	Weber	Wolfgang	Zange	m
321	14	Müser	Angelika	Abel	w

Selektion der Mädchen und Projektion auf die geforderten Angaben.

Ergebnis = $\pi_{\text{Punkte,Name,Vorname}}(\sigma_{\text{Geschlecht=w}}$(InformatikschülerPunkteName))

Punkte	Name	Vorname
14	Müser	Angelika

Aufgaben

Aufgaben zu Operatoren

Aufgabe 1

Gegeben seien drei Relationen mit den folgenden Tupeln:

Besucht

Gast	Bistro
Hans	Uno
Ede	Uno
Ede	Dos
Ede	Chico
Karl	Dos
Karl	Chico
Heini	Uno

Serviert

Bistro	Getränk
Uno	Wasser
Uno	Kaffee
Dos	Kaffee

Mag

Gast	Getränk
Hans	Wasser
Ede	Wasser
Ede	Kaffee
Karl	Kaffee

a) Bilden Sie Serviert × Mag.

b) Bilden Sie *Serviert* $\triangleright\triangleleft$ *Mag*. Welche Informationen beinhaltet diese Relation?

c) Geben Sie alle Bistros aus, die Getränk servieren, die Karl mag. Überprüfen Sie Ihre Operation in der Relationenalgebra anhand des Beispiels.

d) Geben Sie alle Gäste aus, die mindestens ein Bistro besuchen, die auch das Getränk serviert, das sie mögen. Formulieren Sie die Anfrage mit Operationen der Relationenalgebra.

Aufgabe 2

Gegeben seine folgende Relationen (# ist das Zeichen für *Nummer*):

Lieferanten (L#, LName, Status, Stadt)

Teile (T#, TName, Farbe, Gewicht, Stadt)

Projekte (P#, PName, Stadt)

Lieferungen (L#, T#, P#, Anzahl)

Hierbei bedeutet Stadt einmal die Stadt, in der ein Lieferant sitzt, die Stadt, in der das entsprechende Teil hergestellt wird, bzw. die Stadt, in der ein Projekt stattfindet. Lösen Sie die folgenden Aufgaben durch Operationen aus der Relationenalgebra:

a) Finde Sie alle Lieferungen mit Anzahlen zwischen 300 und 750 und geben Sie alle dazu in der Relation Lieferungen verzeichneten Informationen aus.

b) Geben Sie alle Städte aus, in denen Lieferanten sitzen.

c) Geben Sie alle vorkommenden Paarungen TName, Stadt aus.

d) Finden Sie alle schwarzen Teile. Geben Sie ihre Nummer und ihren Namen aus.

e) Finden Sie alle Lieferanten, die in einer Einzellieferung mehr als 150 Teile geliefert haben. Geben Sie ihren Namen aus.

f) Finden Sie alle Teile, die von Lieferanten in London geliefert wurden. Geben Sie davon die Teilenummer (Teilenamen)aus.

g) Finden Sie alle Orte, in denen sowohl Projekte als auch Lieferanten beheimatet sind.

h) Finden Sie alle Projekte, die mindestens einen Lieferanten für das Projekt im gleichen Ort haben. Geben Sie die Projektnummer aus.

i) Finden Sie alle Teile, die der Lieferant Lux geliefert hat. Geben Sie alle Teilinformationen von diesen Teilen aus.

Aufgabe 3

Gegeben sind die folgenden Tabellen:

Tabelle1

A	B	C
4	2	8
2	2	1
6	7	3

Tabelle2

C	D
8	2
3	6

Tabelle3

B	E
5	3
4	4
5	4

Tabelle4

B	C	D
1	2	1
4	2	1
1	2	9

Führen Sie folgende relationalen Operationen durch und stellen Sie die Ergebnistabelle auf! Beschreiben Sie die Aufgabenstellung mittels der behandelten Symbolik!

a) Selektion von Tabelle1 mit der Bedingung B=2
b) Projektion von Tabelle3 auf E
c) Join Tabelle1 und Tabelle2 nach dem gemeinsamen Attribut C
d) (Selektion von Tabelle 1 mit B>C) vereinigt mit (Selektion von Tabelle1 mit A<5)

Aufgabe 4

Gegeben seien folgende Tabellen

GK-Fach 1

Raum	Fach	Lehrer
137	Mathematik	Müller
221	Deutsch	Schmidt
104	Englisch	Lehmann

GK-Fach 2

Raum	Fach	Lehrer
127	Informatik	Müller
104	Englisch	Lehmann
123	Physik	Paulsen
018	Musik	Schmidt

Themen

Nr.	Thema	Klasse
001	Analysis	12/I
002	Klassik	13/1

Bilden Sie

a) GK-Fach 1 $\cap$ GK-Fach 2

b) GK-Fach 1 $\cup$ GK-Fach 2

c) GK-Fach 1 \ GK-Fach 2

d) Themen x GK-Fach 2

e) σ Name = 'Müller'(Schüler)

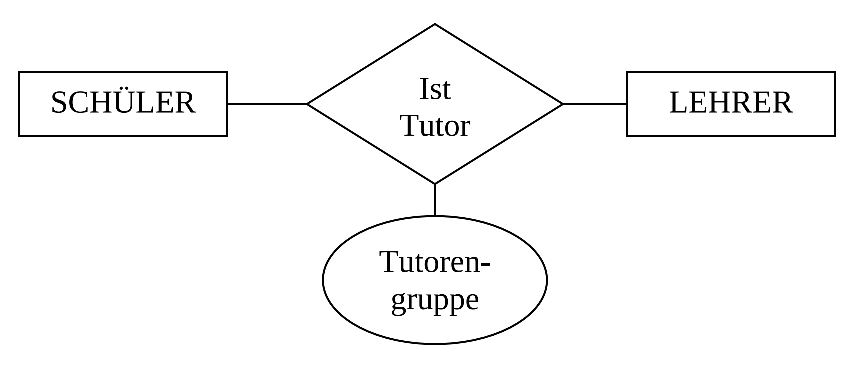
f) π Name(Schüler)

g) π Vorname(σ Name = 'Müller'(Schüler))

h) Join(Schüler, Kurs)

3.4 Übertragung des ER-Modells in das relationale Modell

Bei der Einführung des relationalen Modells haben wir schon gesehen, wie Entitytypen auf Relationenschemata abgebildet werden. Wir wenden uns nun dem Problem zu, Beziehungstypen in das relationale Modell abzubilden. Dazu betrachten wir als Beispiel die *Ist-Tutor*-Beziehung zwischen Schüler und Lehrer:



ER-Diagramm Tutor

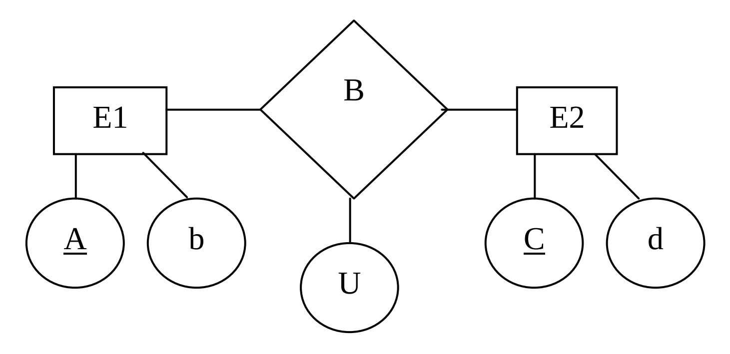
<u>Grundregel</u>:

Jeder Beziehungstyp wird in ein eigenes Relationenschema abgebildet. Die Attribute des Relationenschemas sind die Primärschlüssel der beiden beteiligten Entitytypen zusätzlich der beziehungseigenen Attribute.

Die *Ist-Tutor*-Beziehung wird demnach übersetzt in das Relationenschema:

IstTutor(<u>Schüler-Nr</u>, Lehrer-Nr, Tutorgruppe)

Die Lehrernummer kann kein Schlüssel dieser neuen Relation sein, denn zu jeder Lehrernummer gibt es verschiedene Schülernummern. Da jeder Schüler genau einen Tutor hat, ist hingegen die Schülernummer ein Schlüssel der neuen Relation. Wir sehen, dass die Schlüsselwahl letztlich von der Komplexität der Beziehung abhängt. Im Beispiel ist die Komplexität n:1, also wird der Primärschlüssel der n-Seite zum Primärschlüssel der Beziehungsrelation. Bei einer n:m-Beziehung, beispielsweise *Lehrer unterrichtet Schüler*, bilden die Primärschlüssel der beiden Entityrelationen zusammen den Primärschlüssel der Beziehungsrelation. Bei einer 1:1-Beziehung wählt man einen der beiden Primärschlüssel zum Primärschlüssel der Beziehungsrelation.



ER-Diagramm Tranformationsregeln

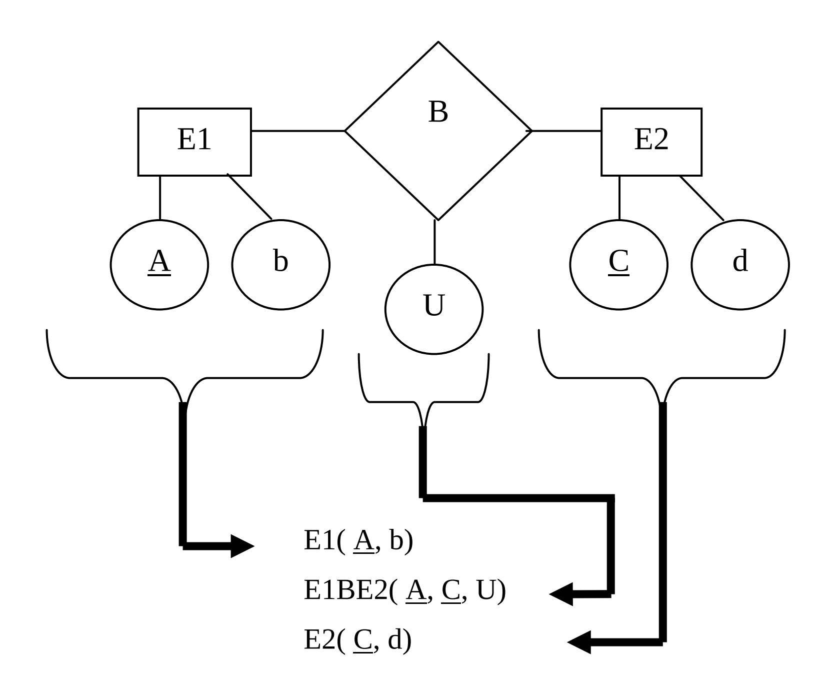
Hier erkennen wir noch mal Allgemein, wie ein ER-Diagramm in das relationale Modell umgewandelt wird. Es sind die beiden Entities E1 und E2 aufgeführt, mit ihren Attributen

A, b und C, d, wobei A und C den Primärschlüssel der jeweiligen Entity darstellen. Die beiden Entities sind über die Beziehungsrelation B verknüpft, die ihrerseits das Attribut U besitzt. Entsprechen ergibt sich nach dem Transformationsschema folgende Tabellenstruktur im relationalen Modell:

E1(A, b)

E1BE2(A, C, U)

E2(C, d)



Präzisierung der Grundregel:

Eine ER-Beziehung zwischen den Entitytypen $E_1(\underline{a_1}, a_2, ..., a_n)$ und $E_2(\underline{b_1}, b_2, ..., b_m)$ mit den Primärschlüsseln a_1 und b_1 sowie den Beziehungsattributen $B(c_1, c_2, ..., c_k)$ wird auf die Beziehungsrelation $BE_{12}(a_1, b_1, c_1, ..., c_k)$ abgebildet. Der Schlüssel der Beziehungsrelation wird nach folgender Tabelle gebildet:

Komplexität der Beziehung	Schlüssel der Beziehungsrelation
1 : 1	a_1 oder b_1
1 : n	b_1
n : m	a_1 und b_1

3.4.1 Optimierungen

Die Grundregel kann immer angewendet werden. Sie liefert unabhängig von der Komplexität der Beziehung stets drei Tabellen für eine binäre Beziehung. Es gibt Sonderfälle, in denen eine ER-Beziehung zwischen zwei Entitytypen auf lediglich zwei Relationen oder gar nur eine Relation abgebildet werden können. Diese Sonderfälle können bei 1:1- und 1:n-Beziehungen auftreten, nicht aber bei n:m-Beziehungen. Diese müssen immer auf zwei 1:n-Beziehungen aufgebrochen werden, da konkrete DBMS keine n:m-Beziehungen direkt darstellen können.

Das bisher benutzte Komplexitätsmaß ist zu grob, um eine Entscheidung darüber treffen zu können, ob man mit weniger Tabellen auskommt. Der Beziehungstyp muß genauer untersucht werden. Dazu betrachten wir die folgenden Relationenschemata etwas genauer:

Schüler(Schüler-Nr, Name,Vorname, Geburtsdatum, Adresse, Schuleintrittsdatum)

IstTutor(Schüler-Nr, Lehrer-Nr, Tutorgruppe)

Lehrer(Lehrer-Nr, Name, Vorname, Titel, Fächer)

In der *Ist-Tutor*-Beziehung können die zwei Tabellen für *Schüler* und *IstTutor* problemlos über ihren gemeinsamen Schlüssel zu einer zusammengefasst werden:

Schüler

Schüler-Nr	Name	Vorname	...
123	Alberti	Hans	
034	Glücklich	Gesine	
321	Müser	Angelika	
111	Weber	Wolfgang	

IstTutor

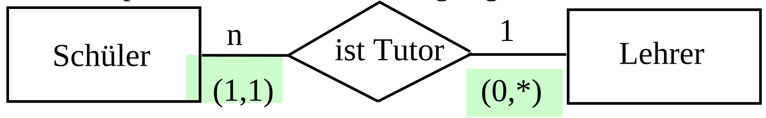
Schüler-Nr	Lehrer-Nr	TutGruppe
123	42	LK/Bio
034	05	LK/Ph
321	37	LK/D
111	42	LK/Bio

Schüler-Tutor

Schüler-Nr	Name	Vorname	...	Lehrer-Nr	TutGruppe
123	Alberti	Hans		42	LK/Bio
034	Glücklich	Gesine		05	LK/Ph
321	Müser	Angelika		37	LK/D
111	Weber	Wolfgang		42	LK/Bio

Die Zusammenfassung funktioniert, weil ein Schüler genau einen Lehrer als Tutor hat. In der Beziehung *IstTutor* kommt jede Schülernummer genau einmal vor. Bei den Lehrernummern ist das anders. Lehrernummern können in der Tabelle keinmal, einmal oder mehrmals vorkommen. Keinmal kommt sie für Lehrer ohne Tutgruppe vor, mehrmals für Lehrer mit Tutgruppe.

Ausgehend von dieser Beobachtung wird die Komplexität einer Beziehung in der Theorie auch durch Angabe von minimaler und maximaler Anzahl, mit der jeweils ein Entitytyp in einer Beziehung vorkommt, angegeben. Ein Schüler kommt in der Beziehung *IstTutor* minimal und maximal einmal vor - Schreibweise (1, 1). Ein Lehrer muss in der Beziehung nicht, kann aber mehr als einmal vorkommen - Schreibweise (0, *). Man sagt auch, dass Schüler *obligatorisch* (zwingend) in der Relation vorkommen, Lehrer hingegen optional. Über die obligatorische Mitgliedschaft an einer Relation müssen in den Geschäftsregeln während des konzeptionellen Entwurfs Aussagen gemacht werden.



Als Beispiel einer nicht obligatorischen Beziehung betrachten wir aus dem ER-Diagramm des Bibliotheksystems die Beziehung:



Ein Ausleiher kann minimal 0, maximal mehr als ein Buch ausleihen - Schreibweise (0, *). Ein Ausleihbuch kann minimal 0, maximal 1-mal ausgeliehen sein - Schreibweise (0, 1).

Ausleihbuch

Inventar-Nr	Buchtyp-Nr	Ausleihzeit
5201000593	3400	180
5201000693	3400	270
5201000793	3500	90
5201000893	3500	180

Ausleiher

Ausleiher-Nr	Name	Vorname
101	Lächa	Roland
204	Möller	Melchior
236	Schulz	Sieglinde
103	Meier	Baltasar

leiht aus

Inventar-Nr	Ausleiher-Nr	Ausleihdatum
5201000593	101	12.07.96
5201000793	204	13.08.96
5201000893	101	29.08.96

Wenn man jetzt versucht, die *leiht-aus*-Tabelle in die *Ausleihbuch*-Tabelle zu integrieren, so entstehen bei nicht ausgeliehenen Büchern Nullwerte in der verbundenen Tabelle:

Anmerkung:

Es gibt noch andere Schreibweisen. Falls Sie auf diese stoßen, dazu hier eine Übersicht:

Unsere Notation	Bedeutung	alternative Notation	Bedeutung
[1, 1]	genau 1	1	1
[0, 1]	0 oder 1	c	can
[1, *]	mindestens 1	m	multiple
[0, *]	keiner oder beliebig viele	mc	multiple can

Ausleihbuch-leiht aus

Inventar-Nr	Buchtyp-Nr	Ausleihzeit	Ausleiher-Nr	Ausleihdatum
5201000593	3400	180	101	12.07.96
5201000693	3400	270	NULL	NULL
5201000793	3500	90	204	13.08.96
5201000893	3500	180	101	29.08.96

Nullwerte sollten bei einem Datenbankentwurf möglichst vermieden werden. Sie sind Ursache unnötiger Komplikationen. Beispielsweise müsste geklärt werden, was das Ergebnis der beiden Selektionen

$\sigma_{\text{Ausleiher-Nr}<200}$ (Ausleihbuch-Ausleiher) und

$\sigma_{\text{Ausleiher-Nr}\geq 200}$ (Ausleihbuch-Ausleiher)

und insbesondere der Mengenvereinigung

$\sigma_{\text{Ausleiher-Nr}<200}$ (Ausleihbuch-Ausleiher) $\cup$ $\sigma_{\text{Ausleiher-Nr}\geq 200}$ (Ausleihbuch-Ausleiher)

ist. Ergibt die Mengenvereinigung die ursprüngliche Tabelle oder nicht?

Die Ursache dieser Komplikationen liegt in den Nullwerten, welche dann in der verbundenen Tabelle entstehen, wenn die n-Seite optional, also nicht obligatorisch an der Beziehung beteiligt ist.

Regel 1 für 1:n-Beziehungen

Ist bei einer ER-Beziehung der Komplexität 1:n zwischen den Entitytypen E_1 und E_2 der Entitytyp E_2 obligatorisch in der Beziehungsrelation B, so können die Relationenschemata von E_2 und B zu einem Schema verbunden werden. Zur Abbildung der Beziehung benötigt man dann nur zwei Tabellen.

$E_1(\underline{a_1}, a_2, ..., a_n)$		$E_1(\underline{a_1}, a_2, ..., a_n)$
$B(\underline{a_1}, \underline{b_1}, c_1, c_2, ..., c_k)$	$\Rightarrow$	
$E_2(\underline{b_1}, b_2, ..., b_m)$		$E_2B(\underline{b_1}, b_2, ..., b_m, a_1, c_1, c_2, ..., c_k)$

Im Sonderfall, dass die Beziehungsrelation keine eigenen Attribute aufweist, reduziert sich die Abbildung einer eine 1:n-Beziehung darauf, dass man das Schlüsselattribut a_1 von E_1 in das Relationschema von E_2 aufnimmt. Ein Join über das in beiden Tabellen enthaltene Schlüsselattribut a_1 kann die Tabellen E_1 und E_2B miteinander verbinden.

Betrachten wir nun Beziehungen der Komplexität 1:1 am Beispiel von Spinden in der Schule.



Nach der Grundregel erhalten wir drei Relationenschemata:

Schüler(<u>Schüler-Nr</u>, Name, Vorname)

Besitzt(*Schüler-Nr*, Spind-Nr) (oder Schüler-Nr)

Spind(Spind-Nr, Standort)

Haben alle Schüler einen Spind, sind also Schüler obligatorisch an der *Besitzt*-Beziehung beteiligt, so kann wie oben die *Besitzt*-Relation mit der *Schüler*-Relation verbunden werden.

Schüler(Schüler-Nr, Name, Vorname, Spind-Nr)

Spind(Spind-Nr, Standort)

Hätte auch nur ein Schüler keinen Spind (optionale Beziehung), so würde bei diesem Schüler ein Nullwert als Spindnummer auftreten. Die beiden Tabellen dürften nicht verbunden werden.

Nehmen wir an, es gibt wenige Spinde und viele Schüler, die Spinde wären also obligatorisch an der Beziehung beteiligt. In diesem Fall könnte man die *Besitzt*-Relation mit der *Spind*-Relation verbinden, ohne dass Nullwerte entstehen:

Schüler(Schüler-Nr, Name, Vorname)

Spind(Spind-Nr, Standort, Schüler-Nr))

Wenn bei einer 1:1-Beziehung beide Entitytypen obligatorisch an der Beziehung teilnehmen, zum Beispiel in der Beziehung *Klasse hat Klassenlehrer*, so können die drei Relationen zu einer einzigen Relation verbunden werden:

```
Klasse(KName,KRaum)
hat(KName,Lehrer-Nr)                          ⇒   KlasseLehrer(Lehrer-Nr, Name, Vorname, KName, KRaum)
Klassenlehrer(Lehrer-Nr, Name, Vorname)
```

Der Entschluss sie dennoch separat aufzunehmen könnte z.B. daher rühren, dass man einen Teil der Daten evtl. für sensibel hält – z.B. das Gehaltsfeld -, während der andere Teil – z.B. die Adresse – für eine breitere Klientel nutzbar sein soll.

Regel für 1:1-Beziehungen

Ist bei einer ER-Beziehung der Komplexität 1:1 zwischen den Entitytypen E_1 und E_2 einer der beiden Entitytypen E’ obligatorisch in der Beziehungsrelation B, so können die Relationenschemata von E’ und B zu einem Schema verbunden werden. Zur Abbildung der Beziehung benötigt man nur zwei Tabellen.

Sind beide Entitytypen obligatorisch in der Beziehungsrelation, so reicht ein gemeinsames Schema für die ER-Beziehung aus:

$E_1(\underline{a}_1, a_2, ..., a_n)$

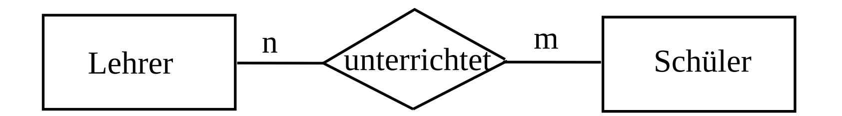
$B (a_1, b_1, c_1, c_2, ..., c_k) \Rightarrow E_1BE_2(\underline{a}_1, a_2, ..., a_n, b_1, b_2, ..., b_m, c_1, c_2, ..., c_k)$

$E_2(\underline{b}_1, b_2, ..., b_m)$

Beziehungen der Komplexität n:m können immer nur auf drei Relationenschemata abgebildet werden.

Die beiden beteiligten Entitytypen werden auf je ein Relationenschema abgebildet, die Grundregel gibt an, wie die Beziehung abgebildet wird. Es ist nicht möglich, zwei der drei entstehenden Tabellen zu verbinden, ohne dass Nullwerte entstehen würden.

Es lohnt sich, die Beziehung zwischen den Entitytyp- und Beziehungsschemata genauer zu betrachten. Die ER-Beziehung *Lehrer unterrichtet Schüler*



ergibt die Relationen

Lehrer(Lehrer-Nr, Name, ...)

unterrichtet(Lehrer-Nr, Schüler-Nr)

Schüler(Schüler-Nr, Name, ...)

mit Tabellen der Art

Lehrer

Lehrer -Nr	Name	...
17	Maier	
68	Schulze	
25	Bauer	

unterrichtet

Lehrer-Nr	Schüler-Nr
17	123
17	034
68	123
68	111
25	111

Schüler

Schüler-Nr	Name	
123	Alberti	
034	Glücklich	
321	Müser	
111	Weber	

1:n n:1

Zwischen Lehrern und den Paaren der *unterrichtet*-Tabelle besteht eine 1:n-Beziehung, denn jeder Lehrer unterrichtet viele Schüler. Diese Beziehung ist auf der n-Seite obligatorisch, weil ein Lehrer, der einen Schüler unterrichtet, per Konstruktion in der *Lehrer*-Tabelle enthalten ist. Analoges gilt für die Beziehung zwischen Paaren der *unterrichtet*-Tabelle und Schülern. Hier besteht eine obligatorische m:1-Beziehung. Man kann also sagen:

Regel für n:m-Beziehungen

Eine n:m-Beziehung kann auf je eine obligatorische 1:n- und m:1-Beziehung aufgeteilt werden.



Regel 3 ist bei der Implementierung eines Datenbanksystems mit einem realen DBMS hilfreich, weil *kommerzielle Systeme lediglich 1:1- und 1:n-Beziehungen unterstützen*!

3.4.2 Sonderfälle

Man muss sich noch Gedanken über die Umsetzung von strukturierten Attributen (Schüleradresse), Mehrfachattributen (Autor beim Buchtyp, Fach beim Lehrer) und *is-a*-Beziehungen machen.

Strukturierte Attribute wie zum Beispiel Adresse können einfach durch Übernahme der Teilattribute Postleitzahl, Wohnort, Straße und Hausnummer in Tabellen abgebildet werden.

Der Versuch, Mehrfachattribute in das Entity-Relationenschema aufzunehmen, scheitert. Entweder man erhält nicht atomare Attributwerte, welche ähnlich wie Nullwerte bei der Selektion Probleme bereiten, oder redundante Wiederholungsgruppen:

nicht atomare Attributwerte Wiederholungsgruppen

Lehrer -Nr	Name	Fach
17	Maier	{M, Ph, Inf}
68	Schulze	{D, E}
25	Bauer	{Spo}

...

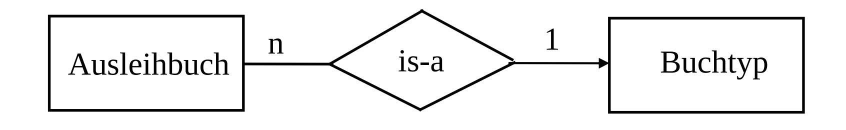
Lehrer-Nr	Name	Fach
17	Maier	M
17	Maier	Ph
17	Maier	Inf
68	Schulze	D
68	Schulze	E
25	Bauer	Spo

Zur Abbildung von **Mehrfachattributen** auf das relationale Modell benötigt man eine zweite Tabelle, welche aus dem Primärschüsselattribut des Entitytyps und dem Mehrfachattribut besteht:

Lehrerfach

Lehrer-Nr	Fach
17	M
17	Ph
17	Inf
68	D
68	E
25	Spo

Is-a-Beziehungen sind stets obligatorische 1:1-Beziehungen (Schüler is-a Ausleiher) oder n:1-Beziehungen (Ausleihbuch is-a Buchtyp). Nach den Regeln 1 und 2 reicht es daher, **in das Relationenschema des spezielleren Entitytyps zusätzlich den Primärschlüssel des allgemeineren Entitytyps aufzunehmen**.



spezieller: Ausleibuch *Allgemeiner*: Buchtyp

→ Primärschlüssel von ’Buchtyp’ (die *Buchtyp-Nr*) zusätzlich in das Relationsschema von ’Ausleihbuch’ aufnehmen

Buchtyp

Buchtyp-Nr	Titel	E-Jahr	Verlag	...
1000	MS Access Benutzerhandbuch	1992	Microsoft	
1001	MS Access Sprachverzeichnis	1992	Microsoft	
1004	MS Access Graph	1993	Microsoft	
1005	MS-Excel 4.0 Schnellübersicht	1992	Markt+Technik	
1006	Multiplan	1986	Hanser	
1007	Quattro Pro für Windows	1993	Borland	
1202	Microsoft Word für Windows	1992	Microsoft	

Ausleihbuch

Inventar-Nr	**Buchtyp-Nr**	Ausleihzeit
5201000593	1004	180
5201000693	1006	270
5201000793	1202	0
5201000893	1007	180

3.4.3 Die Umsetzung der Informatik-Bibliothek ins relationale Modell

Wir wenden nun das im vorherigen Abschnitt dargestellte Verfahren zur Umsetzung eines ER-Modells ins relationale Modell auf unser Beispiel Informatikbücherei (s. Kapitel 2.5) an. Durch Anwendung der Regeln erhalten wir dann die unten dargestellten Tabellen. Der Entity-Typ *Buchtyp* wird in einer Tabelle Buchtyp dargestellt, die is-a-Beziehungen zu *Ausleihbuch* und *Fremdbuch* benötigen keine eigene Tabellen; die beiden Entity-Typen werden durch jeweils eine Tabelle Ausleihbuch und Fremdbuch beschrieben.. Das zusammengesetzte Attribut *Autor* erhält eine eigene Tabelle, ebenso die Beziehung *gehört zu*. Bei der Beziehung *leiht aus* zwischen *Ausleiher* und *Ausleihbuch* handelt es sich um eine 1:n-Beziehung ohne obligatorische Mitgliedschaft. Insofern ist es notwendig, eine Beziehungstabelle Ausleihe zu definieren, die als Schlüssel die Kombination der Attribute *Ausleiher-Nr* und *Inventar-Nr* enthält, dazu noch *Ausleihdatum* als Beziehungsattribut.

Die beiden Entity-Typen *Schüler* und *Ausleiher* werden durch jeweils eine eigene Tabelle dargestellt, die IS-A-Beziehung zwischen ihnen benötigt keine Tabelle. Der Übersichtlichkeit wegen formulieren wir die Tabellenköpfe in der Tupel-Schreibweise:

Buchtyp (Buchtyp-Nr, Titel, Erscheinungsjahr, Verlag, Thema, Kurzbeschreibung)

Autor (Buchtyp-Nr, Autor)

Ausleihbuch (InventarNr, Buchtyp-Nr,Ausleihzeit)

Stichwort (Stichwort)

Ausleiher (Ausleiher-Nr, Name, Vorname)

Gehörtzu (Buchtyp-Nr, Stichwort)

Fremdbuch (Buchtyp-Nr, Ort, Anzahl)

Schüler (Ausleiher-Nr, Klasse, Tutor)

Ausleihe (Ausleiher-Nr, InventarNr, Ausleihdatum)

Die Beziehungen zwischen den Tabellen ergeben sich hier durch die gleichlautenden Attribute (Primärschlüssel). Diese Darstellung ist aus mehreren Gründen unfriedigend, insbesondere werden die Beziehungen und ihre Komplexität nicht deutlich. Es sollte durchaus möglich sein, eine Beziehung zwischen Attributen mit unterschiedlichen Namen in zwei Tabellen herzustellen, wenn die **Wertebereiche** gleich sind. Hier bietet das Programm *Access* eine gute grafische Darstellung, so dass wir darauf zurückkommen, wenn wir die Implementierung in *Access* durchführen.

Es sind aber auch in unserem Relationenschema noch kleine Schwächen, die in einer fehlerhaften Modellierung liegen. Da ist zunächst die Tabelle Stichwort. Alle diese Stichworte tauchen sicher noch in der Tabelle *Gehörtzu* auf, da sogar mehrfach. Wir speichern also Daten doppelt. Diese Mehrfachspeicherung ist insofern notwendig, als sonst das Löschen eines Buchtyps die Löschung der zugehörigen Stichwörter nach sich ziehen würde, obwohl sie sich auch auf andere Bücher beziehen. Eine Möglichkeit wäre es, auf die Tabelle Stichwort ganz zu verzichten. Es bleibt der Nachteil, dass in *Gehörtzu* die doch oft recht langen Stichworte vielfach gespeichert werden. Deshalb erschien es uns sinnvoller, die Tabelle *Stichwort* beizubehalten und mit einem künstlichen Schlüssel *Stichwort-Nr*

zu versehen. In *Gehörtzu* wird dann nur die wenig Speicher erfordernde *Stichwort-Nr* gespeichert.

Ein weiteres Problem liegt in der Speicherung der Ausleihdaten. Wir haben bereits die Beziehungstabelle *Ausleihe* eingeführt. Aber was geschieht bei der Rückgabe eines Buches? Im einfachsten Fall wird der betreffende Datensatz in der Tabelle gelöscht, damit wird auch dem Datenschutz Rechnung getragen. Allerdings sind damit keine statistischen Aussagen mehr über die Häufigkeit von Ausleihen bestimmter Bücher möglich, z.B. um eine sinnvolle Ergänzungsbeschaffung vorzunehmen oder „Ladenhüter" aus der Bibliothek zu entfernen. Hier sind verschiedene Lösungen möglich, die aber nicht weiter diskutiert werden sollen.

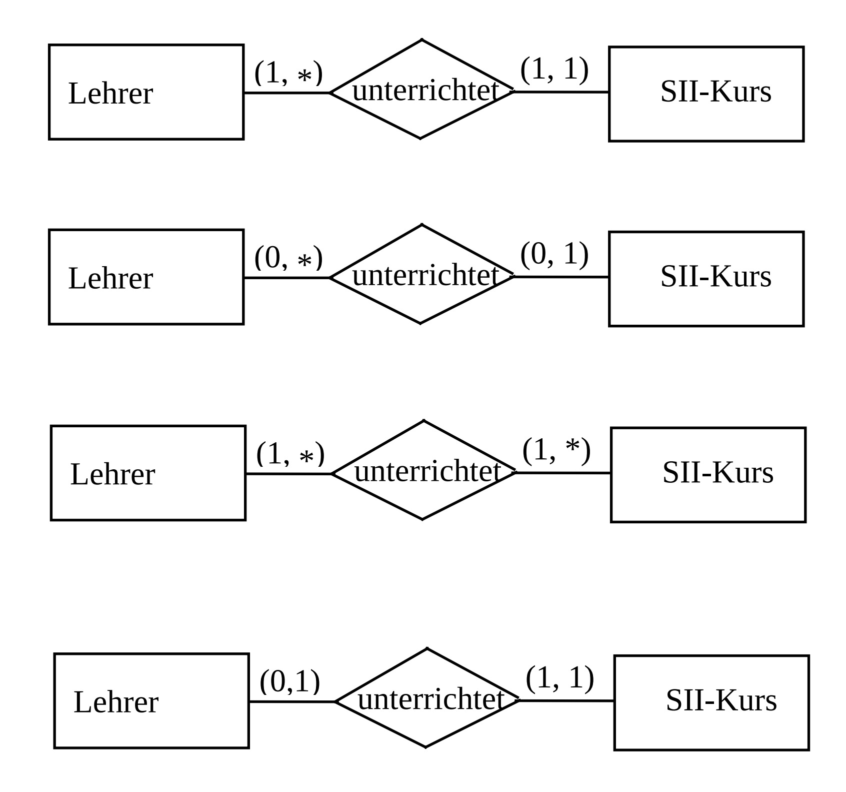
Ebenso lassen sich Anfragen mittels relationaler Operationen formulieren, die auch mehrere Tabellen umfassen können. Es entstehen bei den Anfragen neue Tabellen, die aber nicht als physikalisch in der Datenbank vorhandene Dateien mißgedeutet werden dürfen. Sie entsprechen lediglich einer bestimmten *Sicht* (view) auf die Datenbank, die nur temporären Charakter hat. Wie eine solche Sicht erzeugt wird, hängt von dem benutzten DBMS ab: Es haben sich einerseits standardisierte Abfragesprachen entwickelt, wie z.B. SQL (structured query language), mit deren Hilfe Anfragen formuliert werden. Andererseits bieten moderne Datenbanksysteme wie Access auch einfache Verfahren an, z.B. nach der Methode „query by example" (QBE), die intern auf SQL abgebildet wird.

Aufgaben

Aufgaben zur Umsetzung in das relationale Modell

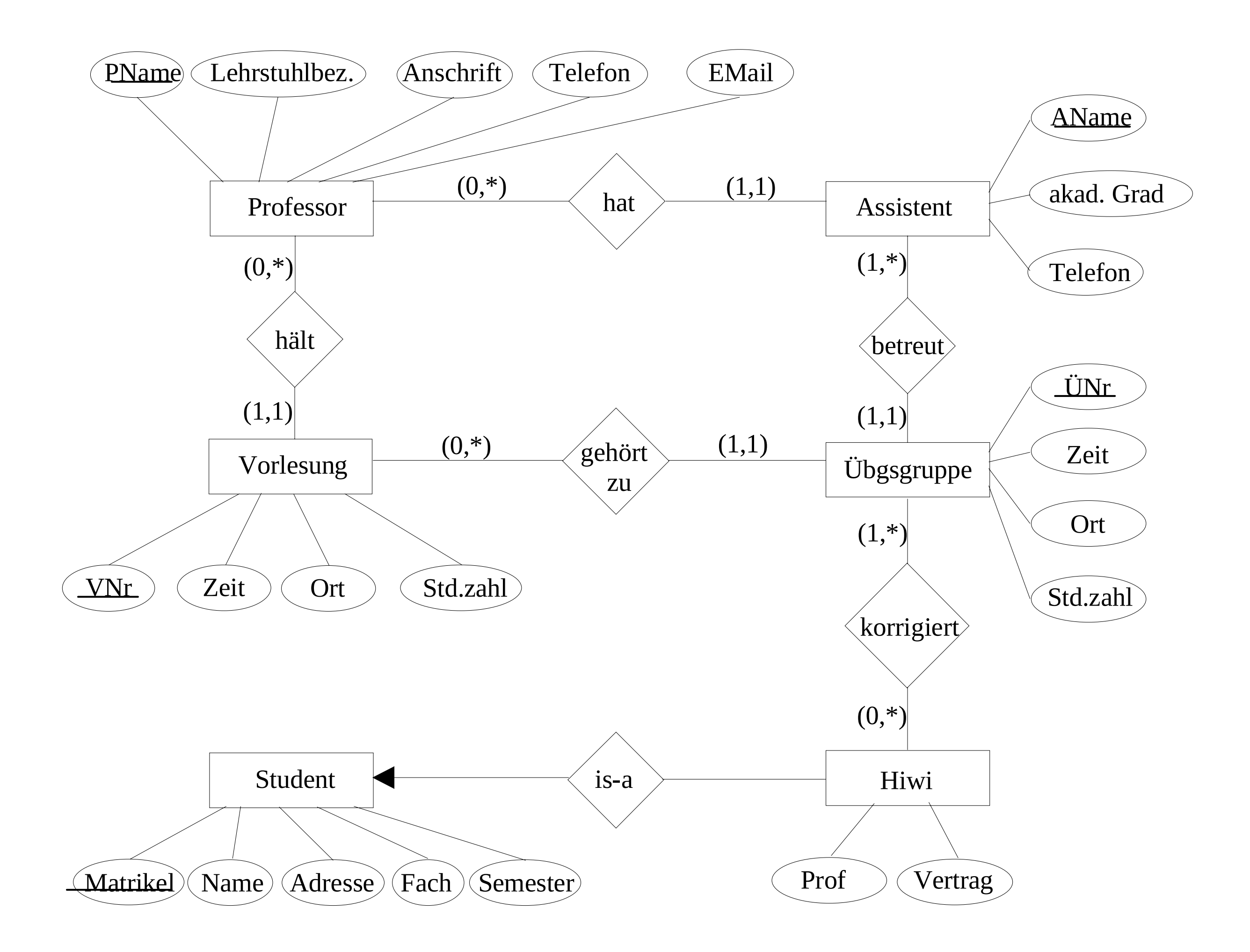
Aufgabe 1

Was bedeuten die folgenden vier Diagramme (Prosa-Erklärung)? Die Attribute wurden zur Vereinfachung weggelassen. Geben Sie jeweils die Komplexität in der Form 1:1, 1:n bzw. n:m an.



Aufgabe 2

Bilden Sie dieses ER-Diagramm in das relationale Modell ab.



Aufgabe 3

Setzen Sie das ER-Diagramm Schule vollständig in ein relationales Modell um, indem Sie alle benötigten Tabellen mit ihren Köpfen angeben, die Primärschlüssel kennzeichnen und durch Pfeile zwischen den Attributen verschiedener Tabellen die Beziehungen verdeutlichen. Im Text wurden schon einige Tabellen eingeführt.

Aufgabe 4

Setzen Sie die unter “Modellierung von Ausschnitten der Wirklichkeit“ dargestellte Aufgabe der Firma Microtec in das relationale Modell um, indem Sie das ER-Modell nach den beschriebenen Verfahren in Tabellen umsetzen!

Aufgabe 5

Setzen Sie das ER-Modell der Spedition Hugo Unbedarft aus „Aufgaben zum ERM“ in ein relationales Modell um. Überlegen Sie dabei, welche Tabellen entstehen müssen und welche Beziehungen sich integrieren lassen.

Anmerkung:
Wenn Sie die Normalisierung erlernt haben kommen sie auf diese Aufgabe zurück und Überprüfen anschließend, inwieweit sich die entstandenen Tabellen in der 3. Normalform befinden! Treten Abhängigkeiten zwischen den Schlüsseln auf?

3.5 Normalisierung

Es gibt noch eine andere Möglichkeit, um auf mehr formalem Weg die benötigten Tabellen mit ihren Beziehungen zu konstruieren. Er führt über die so genannte *Normalisierung.* Codd hat 1972 drei Normalisierungsregeln angegeben die zur 1. Normalform, zur 2. und 3. Normalform einer Relation (Tabelle) führen. Verschiedene Autoren haben später weitere Regeln und Normalformen hinzugefügt, die aber in der Praxis keine große Rolle spielen.

Unter Normalisierung versteht man das Aufteilen aller relevanten Daten in Relationen in der Art und Weise, dass sie am Ende den Normalisierungsregeln entsprechen. Hauptgründe für die Normalisierung sind:

- die Vermeidung unerwünschter Anomalien beim Einfügen, Löschen und Verändern, die zu einer Inkonsistenz der Daten führen.
- die Vermeidung von überflüssiger Information (Redundanz).
- der Zwang zum systematischen Entwurf der Datenbank.
- die bessere Übersichtlichkeit für Benutzer und Programmierer.

Wir wollen uns am Beispiel eines Personalinformationssystems, das die relevanten Daten der Mitarbeiter einer Firma verwaltet, die möglichen Anomalien verdeutlichen. Die drei Tabellen bilden die Entity-Typen *Mitarbeiter* und *Projekt* sowie den Beziehungs-Typ *arbeitet an* ab:

Mitarbeiter

PersNr	Name	Vorname	Beruf	Abteilung	AbtNr
123	Abt	Josef	Buchhalter	Zentrale	4
234	Müller	Lieselotte	EDV-Dozentin	EDV	6
287	Schulz	Eva	Operator	EDV	2

Projekt

ProjNr	Projekt	Leiter
5	DV2000	Müller
5	DV2000	Müller
1	Kosten	Abt

Arbeitet an

ProjNr	PersNr	Telefon	Zeit
5	234	5432	100%
5	287	5433	50%
1	123	65421	100%

Folgende Anomalien können auftreten:

Einfüge-Anomalie:

Eine Mitarbeiterin wird neu eingestellt. Da sie sich erst einarbeiten muss, arbeitet sie noch an keinem Projekt mit. Ihre Telefonnummer kann nicht gespeichert werden.

Lösch-Anomalie:

Das Projekt 5 ist abgeschlossen und wird aus der Datenbank entfernt. Dabei gehen auch die Informationen über die Telefonnummern verloren.

Änderungs-Anomalie:

Eine Mitarbeiterin, die gleichzeitig Projektleiterin ist, ändert durch Heirat ihren Namen. Dieser wird in der Tabelle MITARBEITER geändert, verbleibt aber in der Tabelle PROJEKT unverändert.

Um solche Anomalien und Redundanzen zu vermeiden, werden die Normalisierungsregeln angewandt. Man geht dabei von allen relevanten Daten aus und stellt sie in einer Tabelle zusammen. In einer unnormalisierten Tabelle gibt es meist Wiederholungsgruppen (Mehrfachattribute) und Redundanzen.

PersNr	Name	AbtNr	ProjNr
123	Müller	5	6,3

Deshalb müssen Wiederholungsgruppen in eine neue Relation ausgegliedert werden:

PersNr	Name	AbtNr
123	Müller	5

PersNr	ProjNr
123	6
123	3

Die so bereinigte Relation befindet sich dann in der *1. Normalform.*

1. Normalform

Eine Relation befindet sich in der ersten Normalform, wenn an allen Kreuzungspunkten von Zeilen und Spalten nur atomare Werte auftreten.

Ein Datensatz muss durch den Primärschlüssel eindeutig identifizierbar sein.

Anmerkung*:*

Es gibt in der 1.Normalform auch *keine Wiederholungsgruppen* mehr. Diese können formal durch mehrere Zeilen aufgelöst werden oder gleich – was geschickter ist – durch Ausgliederung in eine eigene Tabelle, verbunden über den Primärschlüssel der ursprünglichen Tabelle.

Beispiel:

Rgnr	Name	Vorname	Straße	Nr	Plz	Ort	**ArtNr1**	**ArtBez1**	***ArtNr2***	***ArtBez2***
1	Peters	Klaus	Xyz Gasse	1	33333	Testingen	12345	Wasser	45678	Limo
2	Schulze	Fried	Abc Gasse	4	22222	Mohingen	99999	Bier	44444	Sekt

wird aufgelöst, in dem mehrere Zeilen für die Wiederholungsgruppe spendiert werden

Rgnr	Name	Vorname	Straße	Nr	Plz	Ort	**ArtNr**	**ArtBez**
1	Peters	Klaus	Xyz Gasse	1	33333	Testingen	12345	Wasser
3	Peters	Klaus	Xyz Gasse	1	33333	Testingen	45678	Limo
2	Schulze	Fried	Abc Gasse	4	22222	Mohingen	99999	Bier
4	Schulze	Fried	Abc Gasse	4	22222	Mohingen	44444	Sekt

oder besser durch eine eigene Tabelle, in der die Wiederholungsgruppe sowie der Primärschlüssel der Ursprungstabelle mit aufgenommen werden.

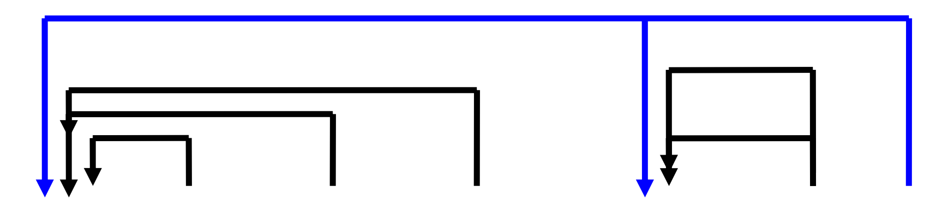
Rgnr	Name	Vorname	Straße	Nr	Plz	Ort
1	Peters	Klaus	Xyz Gasse	1	33333	Testingen
2	Schulze	Fried	Abc Gasse	4	22222	Mohingen

Rgnr	ArtNr	ArtBez
1	12345	Wasser
3	45678	Limo
2	99999	Bier
4	44444	Sekt

Dabei ist die neue Tabelle über den Primärschlüssel *Rgnr* mit ihrer Ursprungstabelle verbunden. Zu beachten ist, dass allerdings nun der alte PS, sprich *Rgnr*, nicht mehr eindeutig ist, denn es können ja mehrere Artikel in ein und derselben Rechnung erscheinen. Entsprechend muss zu hier zu einem zusammengesetzten Primärschlüssel übergegangen werden!

Die folgende Tabelle eines Personalinformationssystems befindet sich in der 1. Normalform (es sind der Übersichtlichkeit wegen nicht alle Attribute dargestellt):

Firma



PersNr	Name	AbtNr	Abteilung	ProjNr	Projekt	Zeit
123	Müller	5	EDV	6	Novell	50%
123	Müller	5	EDV	3	DV2000	50%
876	Schulze	3	Personal	3	DV2000	100%

Um die weiteren Normalisierungsschritte vornehmen zu können, sind die Begriffe *funktional abhängig* und *transitiv abhängig* zu klären. Sie beziehen sich auf die Abhängigkeit zwischen den Schlüssel- und den Nicht-Schlüssel-Attributen.

In einer Relation ist das Attribut B vom Attribut A *funktional abhängig*, wenn zu jedem Wert von A genau ein Wert von B gehört. In unserer Tabelle sind die Attribute *Name, AbtNr, Abteilung* funktional abhängig vom Attribut *PersNr*. Dieses Attribut eignet sich daher als Schlüssel für eine Tabelle, die nur diese Daten enthält. Das Attribut *Projekt* hängt funktional vom Attribut *ProjNr* ab, das deshalb Schlüssel einer weiteren Tabelle werden kann. Das Attribut *Zeit* hängt dagegen nur von der Kombination der beiden Schlüsselattribute *PersNr* und *ProjNr* ab, nicht dagegen von einem Teil dieses zusammengesetzten Schlüssels.

2. Normalform

Eine Relation befindet sich in der 2. Normalform, wenn sie sich in der ersten befindet und jedes Nicht-Schlüssel-Attribut funktional abhängig ist vom Gesamtschlüssel, nicht dagegen von Schlüsselteilen.

Ferner sollte jede Tabelle nur Daten aus einem Objektbereich beinhalten.

Objektbereiche wie z.B. Personaldaten und Abteilungsdaten werden damit entzerrt.

Nicht-Schlüsselattribute, die nur von einem Teil des PS abhängen, werden mit diesem als Attribut in eine neue Tabelle ausgelagert, so dass das Kriterium für die 2.Normalform damit erfüllt ist.

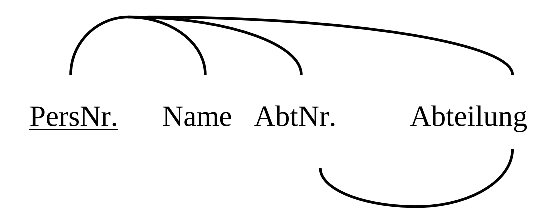
Die Aufspaltung der obigen Tabelle in 3 Tabellen überführt diese in die 2. Normalform. Die verwendete Tupel-Schreibweise ist uns schon bekannt (s. Kapitel 3.1) und erleichtert den Überblick:

Mitarbeiter(PersNr,Name,AbtNr,Abteilung)

Projekt(ProjNr,Projekt)

Arbeitet an(PersNr,ProjNr,Zeit)

Ohne die Tabellen nochmals mit Werten hinzuschreiben, macht man sich leicht klar, dass es trotzdem noch Datenredundanz gibt. So müsste bei jedem Mitarbeiter, der eingestellt wird, neben der *AbtNr* auch *Abteilung* gespeichert werden. Das kann natürlich auch zu Einfüge- und Änderungsanomalien führen. Wie man leicht einsieht, liegt der Grund dafür in der funktionalen Abhängigkeit der Nichtschlüsselattribute *Abteilung* vom ebenfalls Nichtschlüsselattribut *AbtNr.* Diesen Sachverhalt kann man wieder mathematisch so ausdrücken: Das Nicht-Schlüssel-Atttribut *Abteilung* ist nur *transitiv abhängig* vom Schlüssel-Attribut *PersNr,* weil es nur über den Umweg *AbtNr* von *PersNr* abhängig ist.



Man kann das auch so formulieren: *Abteilung* hängt funktional von *AbtNr* ab und *AbtNr* wieder von *PersNr.* Wenn B von A abhängt und C von B, dann hängt C über B indirekt von A ab, eben transitiv. Im Gegensatz zur transitiven Abhängigkeit wollen wir von *direkter Abhängigkeit* sprechen, wenn es keinerlei mögliches „Zwischenglied“ für die Abhängigkeit gibt.

Wenn wir die transitive Abhängigkeit beseitigen wollen, müssen wir die Tabelle MITARBEITER zerlegen, es entstehen zwei Tabellen. Diese befinden sich dann in der 3. Normalform nach Codd:

Mitarbeiter(PersNr,Name,AbtNr)

Abteilung(AbtNr, Abteilung)

3. Normalform

Eine Relation befindet sich in der 3. Normalform, wenn sie sich in der 1. und 2. Normalform befindet und keine funktionalen Abhängigkeiten zwischen Nicht-Schlüssel-Attributen existieren. Anders ausgedrückt: Die Relation darf keine transitiven Abhängigkeiten aufweisen.

Nicht-Schlüsselattribute, die von anderen Nicht-Schlüsselattributen abhängig sind, werden mit diesen als Attribut in eine neue Tabelle ausgelagert, so dass das Kriterium für die 3.Normalform damit erfüllt ist.

Damit haben wir zumindest in unserem Beispiel redundante Daten entfernt und die erwähnten Anomalien vermieden. Es lassen sich jetzt komplexere Beispiele konstruieren, in denen doch wieder Probleme auftreten, obwohl sich die Tabellen in 3NF befinden. Dazu gibt es weitere Normalisierungsmethoden, unter anderem die so genannte Boyce-Codd-Normalform, die noch über die 3NF hinausgehen.

Wir fassen die Erkenntnisse über den systematischen Normalisierungsprozess nochmals kompakt in einer Grafik zusammen.

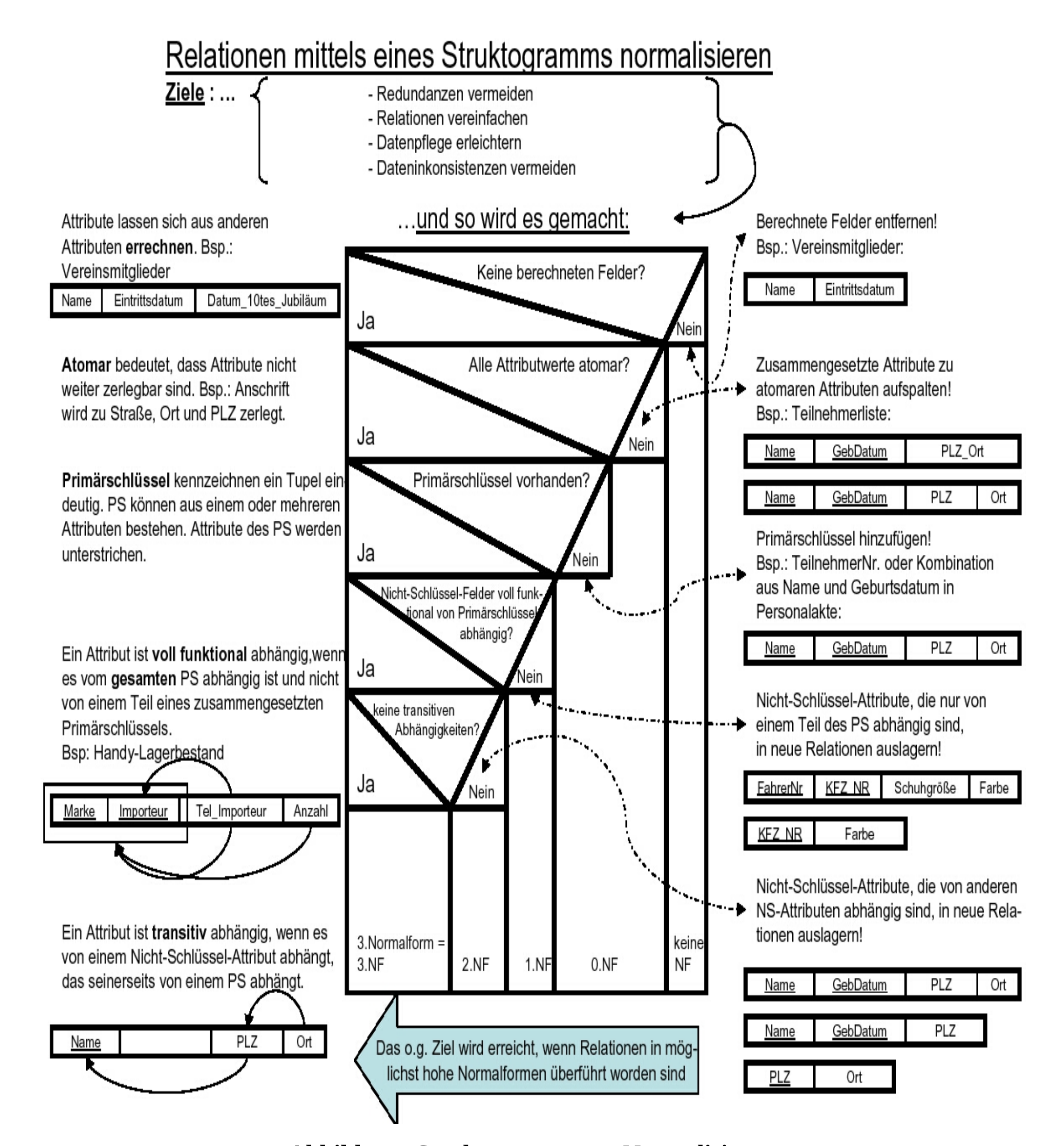


Abbildung: Struktogramm zur Normalisierung

Reflexion Datenbankdesign

Mit unseren jetzigen Kenntnissen können wir noch mal der Frage nach dem allgemeinen prinzipiellen Vorgehen beim Datenbankentwurf nachgehen und gelangen zu etwa folgender Vorgehensweise:

1. Bestimmung der Aufgaben der Datenbank: WAS soll sie überhaupt leisten?
2. Tabellen / Entitäten / Objekte bestimmen
3. Felder (Attribute) der Tabellen bestimmen
4. Beziehungen definieren
5. Entwurfsverfeinerung mittels Normalformen
6. Umsetzung mit einem RDBMS

Zu 1: Es hilft oft, sich typische Fragen zu notieren, die die Datenbank einmal beantworten können muss, z.B.

- Wer ist der Lieferant des meistverkauften Artikel?
- Wie groß war der Umsatz im Mai vorigen Jahres und diesen Jahres?
- Welcher Sachbearbeiter ist für den Kunden X zuständig?

Dann sollte man vorliegende Dokumente beschaffen und Analysieren, die den abzubildenden Gegenstandsbereich der Datenbank betreffen, z.B. Berichte, Rechnungen, Auftragsbestätigungen usw. So kann man fundiert Objekte und ihre Attribute identifizieren und gewinnt ferner ein schärferes Bild vom Gegenstandsbereich, der dem Datenbankentwickler ja häufig fremd ist!

Zu 2: Vermeiden von Redundanzen durch Komponentenbildung. Ergibt sich oft auf natürliche Art und Weise, in dem man das was logisch zusammen gehört auch als ein Objekt betrachtet und auf eine Tabelle abbildet, z.B. Personendaten, Bestelldaten, Produktdaten…

Aufgaben

Aufgaben zur Normalisierung

Aufgabe 1

Ein Hochschulberater berät Studenten, die alle im Wohnheim der Universität leben und alle das gleiche Hauptfach studieren. Aus Besprechungen mit Studenten will der Berater eine kleine Datenbank zur Unterstützung der Beratung entwickeln. Er legt folgende Attribute und Regeln fest:

SNr	Studentennummer, ganze Zahl , eindeutiger Schlüssel für Studenten
SName	Name des Studenten, nicht eindeutig
ZNr	Zimmernummer ganzzahlig; jeder Student hat ein Zimmer, das aber von mehreren Studenten bewohnt werden kann
TNr	Telefonnummer des Studenten; sie ist für alle Zimmerbewohner gleich
Kurs	Identifikationsnummer der Kurse, die ein Student belegt oder bereits abgeschlossen hat
Semest	Semester, in dem ein Kurs abgeschlossen wurde; ein Student darf den gleichen Kurs in einem späteren Semester wiederholen
Note	Note eines abgeschlossenen Kurses

Der Berater möchte folgende Daten in der Datenbank speichern (Auszug):

SNr	SName	ZNr	TNr	Kurs	Semest	Note
3215	Jonas Mike	120DH	2136	MAT122	W88	1,4
				PHY120	S88	2,5
				WIW330	W89	3,1
				MAT122	S89	1,2
3456	Schmidt Klaus	237VH	3127	MAT122	W87	3,2
				MAT130	S87	2,9
4576	Neider Paul	120DH	2136	PHY230	W88	2,8
				MAT120	S88	2,1

a) Erstellen Sie ein Diagramm zum ER-Modell für diese Daten.

b) Machen Sie aus der obigen Tabelle eine gültige Relation in der 1. Normalform. Welche Anomalien können dabei auftreten?

c) Erstellen Sie anhand des ER-Modells und der Umsetzungsregeln das relationale Modell. Begründen Sie, inwieweit die Tabellen die 3. Normalform erfüllen.

d) Geben Sie die relationalen Operationen an, um Kurslisten auszugeben.

Aufgabe 2

Ein kleines Belegkrankenhaus stellt auf EDV um. Die Patientendaten enthalten die folgenden Attribute in Tupelschreibweise:

PATIENT = (Name,Gebdatum, Adresse, Blutdruck, Datum, Diagnose, Therapie, Datum, Diagnose, Therapie,...,Belegung)

Anmerkung: *Systol*: zusammenziehen; Kontraktion eines Hohlorgans

Diastol: erschlaffen; Erschlaffen eines muskulären Organs zwischen 2 Systolen

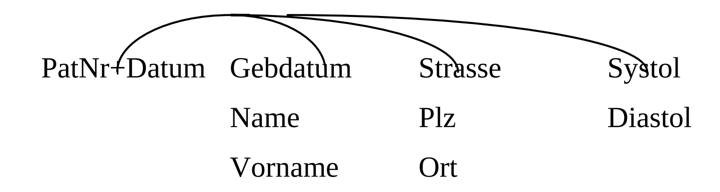
Beides bezieht sich auf den Blutdruck.

Erklären Sie anhand der folgenden Abbildungen den Gang der Normalisierung und begründen Sie die einzelnen Schritte. Bilden Sie dabei Klassen , gliedern Sie also die Attribute unter Oberbegriffen.

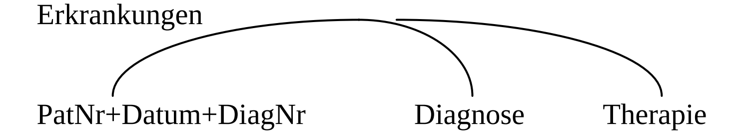
1. Normalform



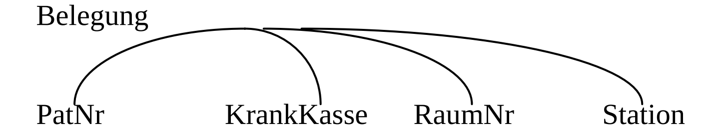
Patient



Erkrankungen

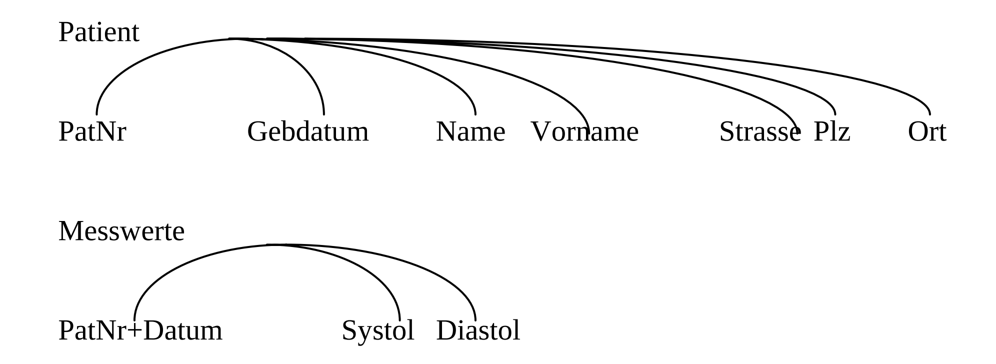


Belegung



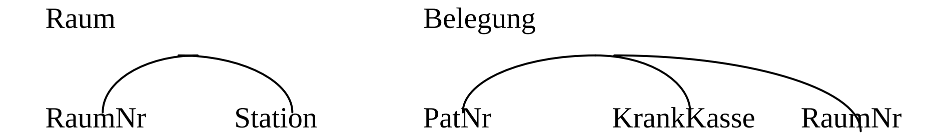
2. Normalform

Die Datengruppe PATIENT wird weiter zerlegt:



3. Normalform

Die Datengruppe BELEGUNG wird weiter zerlegt:



Geben Sie für die 3. Normalform die entsprechenden Tabellen im relationalen Modell an!

Aufgabe 3

Sie haben die Aufgabe der Spedition von Hugo Unbedarf bereits gelöst und sich auch schon über die Tabellen und Verbindungen Gedanken gemacht, bevor Sie dieses Thema explizit erlernt haben. Schauen Sie sich jetzt die Aufgabe noch mal an und lösen Sie sie formal im Lichte Ihres jetzigen Kenntnisstands.

Aufgabe 4

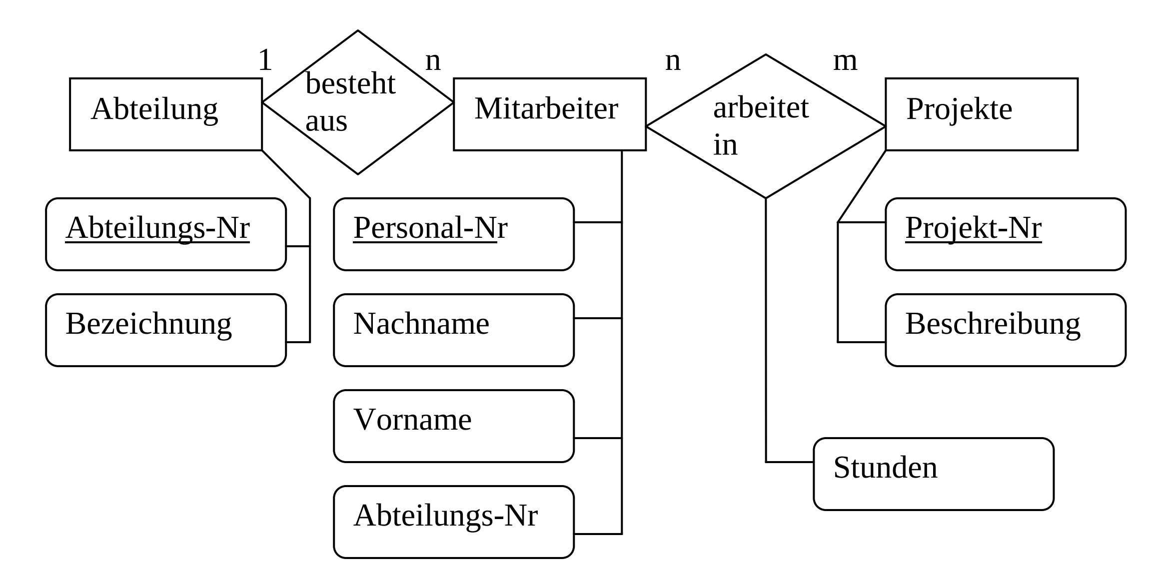
Eine Tabelle mit Lehrerdaten sei wie folgt formuliert

Nr	**Nachname**	**Vorname**	**Amtsbez.**	**Besoldungsgruppe**	**Klasse**
1	Bavaria	Eusebia	StRin	A13	11WA, 13ET
2	Bachmann	Hanna	StRin	A13	11WB
3	Kalkulus	Carl-Johann	StD	A15	12SC
4	Spike	Moses	OStR	A14	12GB

Analysieren Sie die Tabelle. Welche Schwächen fallen auf? Normalisieren Sie sie anschließend.

Aufgabe 5

Gegeben sei folgendes ERM.



Geben Sie die sich ergebende relationale Tabellenstruktur in der Kompaktschreibweise “Tabellenname(*PS*, Attribute)“ an. Leiten Sie dabei Ihre Lösung ausführlich her und dokumentieren Sie die einzelnen Schritte.

Aufgabe 6

Sie importieren Daten aus einer Excel-Tabelle und erhalten die folgende Struktur. Normalisieren Sie sie bis zur 3. Normalform. Dokumentieren Sie dabei ausführlich Ihre Schritte mit der entsprechenden Begründung (Definition der Normalformen und ihre Folgerungen).

PersNr	Nachname	Vorname	Abteilung	Projekt	Stunden
1	Lorenz	Christian	Einkauf	Verkaufsanalyse	198, 201
2	Baumann	Peter	Verkauf	Optimierung	120, 189, 43
3	Petersen	Anna	Personal	Weiterbildung	120

Aufgabe 7

Reflektieren Sie nochmals über das systematische Vorgehen während des Normalisierungsprozesses. Dazu studieren Sie vorliegenden Text und ergänzen die sich ergebenden Tabellen.

Anmerkung: Es gibt mehrere denkbare Lösungen, also mehrere Tabellenstrukturen. Falls Ihre Idee nicht in die vorgesehenen Tabellen passt ergänzen Sie einfach die Spaltenanzahl, fragen sich aber kritisch, ob Ihre Lösung den gestellten Anforderungen genügt. Das primäre Ziel der Aufgabe ist die nochmalige intensive Reflexion des systematischen Normalisierungsprozesses.

Ein Bücherwurm Karlo möchte Ordnung in sein ganz privates Bücherchaos bringen, da ihm die Unmengen an über die Jahre erworbenen Büchern mittlerweile ein wenig über den Kopf gewachsen sind. Entsprechend entschließt er sich spontan eine Datenbank (DB) dafür zu

entwickeln. Seine Freundin studiert gerade fleißig Informatik und kommt am Wochenende vorbei, da möchte unser edler Gesell’ natürlich ein wahres Kunstwerk mit allen Schikanen vorweisen können.

Als Regeln für gutes Datenbank-Design findet er folgende Weisheiten

- Tabellen sollten keine redundanten (sich wiederholende) Daten enthalten. Wenn in einer Tabelle immer wieder dasselbe eingegeben werden muss, stimmt etwas nicht.
- Bei der Konzeptentwicklung der DB (konzeptuelles Modell) bietet sich z.B. Excel an, da man damit leicht die Tabellen und ihre Attribute visualisieren kann, ohne auf die spezifischen Restriktionen eines echten DBMS achten zu müssen

Diese Regeln sind formalisiert als Normalisierung bekannt.

Als erster Schritt im DB-Design ist der Weltenausschnitt zu modellieren, den man gerade betrachtet. Bei einer Bücherdatenbank sind entsprechend die Attribute zu ermitteln, die in der DB verwaltet werden sollen. Es wären dies beispielsweise

- Titel (title)
- Verlag (publName)
- Veröffentlichungsjahr (year)
- Autoren (authorX)

Bei der Ermittlung dieser Attribute helfen Realweltdaten wie Karteikarten, Akten usw. Eine handschriftliche Bücherliste von Karlo hat beispielsweise folgende Struktur:

title	publName	year	author1	author2	author3
Linux	Addison-Wesley	2004	Meyer		
Web	Apress	2003	Schulz	Einstein	
Mathe	New Riders	2006	Schmitz		
Physik	Einstein-Verlag	1901	Newton	Schrödinger	Heisenberg
Lyrik	Federweiß	2006	Schulz		

Nun ja, also genau genommen hatte bei der Physik auch noch Max Planck seine Finger im Spiel, aber es war halt nur Platz für drei Autoren… Nicht sehr schön. Die Festlegung auf drei Autoren ist willkürlich, unflexibel und schlichtweg unschönes Design. Auch die Lösung alle Autoren in ein einziges Feld authors zu schreiben

authors
Newton, Schrödinger, Heisenberg, Planck

ist nicht so das wahre, da Karlo gehört hat man solle pro Attribut, also pro Spalte in einer Tabelle, nur *atomare Werte*, also ein Datum statt mehrerer, verwalten.
Nun denn, als Designkriterien für diese erste Hürde gelten folgende Weisheiten , die der *ersten Normalform*:

- Nur atomare Werte pro Spalte.
- Spalten mit gleichartigem Inhalt müssen entfernt werden.
- Für jede Gruppe zusammengehöriger Daten muss eine eigene Tabelle gebildet werden.
- Jeder Datensatz muss durch einen sog. Primärschlüssel eindeutig identifiziert werden.

Die dritte Regel ist schon erfüllt, da momentan in jeder Spalte nur Daten gesammelt werden, die auch zu einem Buch gehören. Regel 1 ist gleichfalls erfüllt, da das Problem der Mehrfachdaten pro Attribut von Karlo durch die drei Autorenspalten gelöst wurde, auch wenn sich dadurch Probleme mit Regel zwei eingefangen hat. Regel 4 ist leicht zu befolgen. Es wird einfach eine fortlaufende Nummer eingeführt (AutoWert), die *titleID*, und als Primärschlüssel verwendet. Diese Nummer identifiziert dann das Buch mit all seinen Attributen eindeutig. Entsprechend ergibt sich die Tabelle nun wie folgt:

Prima, alle Kriterien der ersten Normalform sind erfüllt! Selbst das Autorenanzahlproblem wurde gelöst. Allerdings mag es doch verdächtig anmuten alle Daten für n-Autoren n-mal vorrätig zu halten. Das kann nicht der Weisheit letzter Schluss sein.
Eine Tabelle wird in die *zweite Normalform* umgeformt, in dem

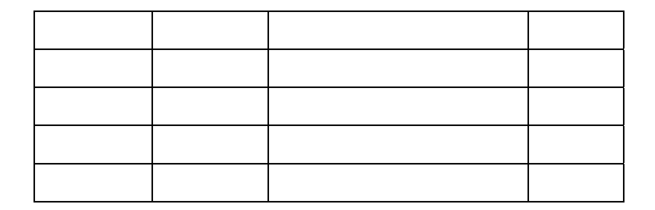
- Eine Tabelle in erster Normalform ist.
- Immer dann, wenn sich Inhalte in Spalten wiederholen, müssen die Tabellen in mehrere Teiltabellen zerlegt werden.
- Die Tabellen müssen durch sog. Fremdschlüssel miteinander verbunden werden.

In obiger Tabelle wiederholen sich Daten auf Grund der Autoren. Eine Auslagerung derselbigen ergibt folgende Tabellenstruktur:

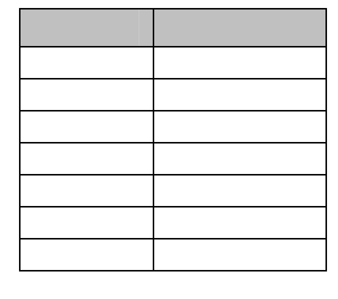
authID	titleID	author
1	1	Meyer
2	2	Schulz
3	3	Schmitz
4	4	Newton
5	4	Schrödinger
6	4	Heisenberg
7	4	Planck
8	5	Schulz

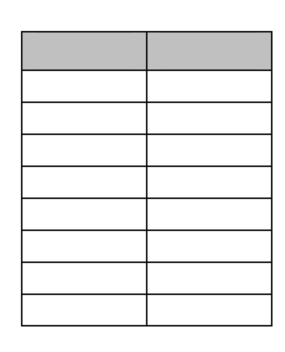
Schon besser. Allerdings kommt 'Schulz' als Autor immer noch mehrmals vor. Dies tritt immer dann auf, wenn ein Autor mehr als ein Buch geschrieben hat. Als Lösung käme eine weitere Aufteilung der Tabelle in Frage:

titles



authors *rel_title_author*





Letztere Tabelle muss von Rechts nach Links gelesen werden. Sie gibt also an, welche Bücher ein Autor – identifiziert über seine eindeutige authID – geschrieben hat. Vom Buch in einer Anfrage geht es also über die titleID zur authID, welche schließlich zu den Autoren des Buches führt. Die Tabelle rel_title_author hat übrigens einen *zusammengesetzten Primärschlüssel*, der aus den beiden Attributen besteht. Er identifiziert jedes Tupel eindeutig, daher ist ein zusätzlicher Primärschlüssel nicht nötig.
Es gibt noch eine weitere wichtige Normalform, die *dritte Normalform*. Diese löst indirekte (*transitive*) Abhängigkeiten auf. Ist A von B abhängig und B von C, so ist A transitiv von C abhängig, was nicht erlaubt ist, wenn eine Tabelle der dritten Normalform genügen soll. Ein typisches Beispiel ist die Postleitzahl und der Ort. In einer Mitarbeitertabelle werden die Personalstammdaten inklusive Wohnort erfasst. Eine MitarbeiterID (A) identifiziert den Mitarbeiter eindeutig, auch die zugehörige PLZ (B). Der Wohnort (C) ist aber abhängig von der PLZ (B) und damit besteht transitive Abhängigkeit zwischen MitarbeiterID und Wohnort. Entsprechend werden PLZ und Ort aus der Tabelle in eine eigene ausgegliedert.

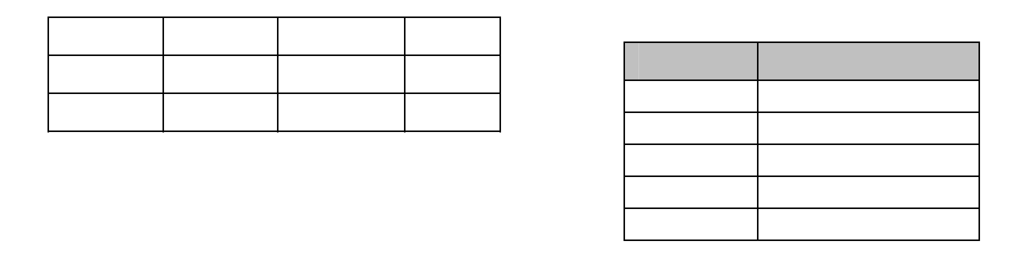
PersID	Name	…	PLZ	Ort
1			33433	Marl
2			33453	Gelsenkirchen
3			2432	New York

→

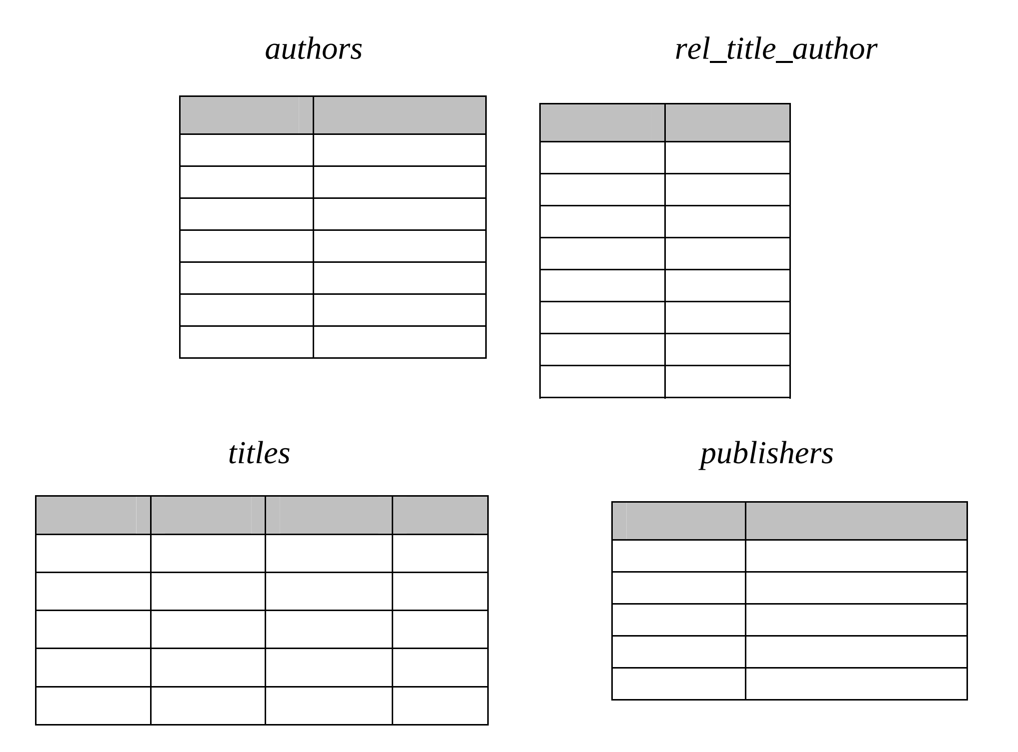
PersID	Name	…	PLZ
1			33433
2			33453
3			2432

PLZ	Ort
33433	Marl
33453	Gelsenkirchen
2432	New York

Entsprechend reagiert Karlo und findet heraus, dass die Spalte *publName* in der *title*-Tabelle betroffen ist. Die Liste der Verlage und die Liste der Buchtitel sind voneinander abhängig und sollten daher getrennt behandelt werden. Es ist zwar nötig und sinnvoll zu jedem Buch den Verlag zu vermerken, aber es reicht doch ein kurzer Schlüssel statt immer der vollständigen Daten (hohen Fehleranfälligkeit und große Speicherplatzverschwendung, was zudem den Zugriff auf die Daten verlangsamt). Dies ist der Hintergrund für die dritte Normalform.
Gesagt, getan. Karlo macht folgendes daraus.



Die Tabellen *rel_title_author* und *authors* änders sich in der dritten Normalform nicht mehr. Die gesamte Bücherdatenbank besteht nun aus drei Tabellen.



Die Normalformen formalisieren demnach zwei Designprinzipien:

- Fasse in einer Tabelle nur das zusammen, was zusammen gehört.
- Vermeide Redundanzen und Abhängigkeiten durch Separierung in eine eigenen Tabelle.

Erweiterungen:

Die Datenbank könnte noch leicht um einige weitere – realistische – Features erweitert werden. So könnte die *titles*-Tabelle noch weitere Felder erhalten, wie etwa einen *Untertitel* oder einen *Kommentar*. Eine weitere languages-Tabelle könnte eine Liste aller Sprachen enthalten, in denen das Buch erschienen ist. Ein Verweis darauf könnte über ein *langID*-Feld in der Tabelle *titles* erfolgen. Eine Tabelle *categories* könnte eine Liste der Buchkategorien enthalten, die hierarchisch gegliedert ist, z.B. 'computer books → databases' und 'computer books → programming', 'computer books → programming → java' usw. Erfasst werden könnte diese Kategorie-Datailtabelle über ein Feld *catID* in der Tabelle *titles*.

Exkurs: Hierarchien

Categories ordnet Bücher in verschiedene Kategorien, die hierarchisch strukturiert sind. Dabei verweist *parentID* auf die übergeordnete Ebene.

- All Books
  - Children books
  - Computer books
    - Databases
      - Object oriented db
      - Relational db
      - Sql
    - Programming
      - Pearl
      - Php

→

catID	catName	parentID
1	computer books	11
2	databases	1
3	programming	1
…		
11	all books	11

Aufgabe 8

Eine Möbelfirma stellte vom Karteikartensystem

Artikelbezeichnung: **Gabriele**		
Artikelart: **Schreibtisch**	Artikelnummer: **1000**	
Bestand:	Mindestbestand: **5**	Höchstbestand: **10**
Einkaufspreis:	Verkaufspreis:	
Lieferant: **Tranel GmbH**	Name2: **Büromöbel**	
Lieferer_Nr: **102**	Anschrift: **Bechemstr. 67, 47058 Duisburg**	

auf Excel um und möchte nun – mit steigendem Datenaufkommen – eine echte Datenbank nutzen. Es gibt sich nach dem Import folgende Tabellenstruktur. Normalisieren Sie sie.

Artikel_Nr	Artikelart	Artikel_Bez	LiefererNr	Name1	Name2	Anschrift
1000	Schreibtisch	Gabriele	102	Tranel GmbH	Büromöbel	Bechemstr., 33242 Duisburg
1001	Schreibtisch	Modern	101	Büromöbel AG	Büroeinrichtungen	Gutachtstr. 342, 12345 Bremen
1002	Schreibtisch	Exklusiv	102	Tranel GmbH	Büromöbel	Vogtweg 34, 32453 Verlin
1003	Büroschrank	Elegant	100	Wagner GmbH	Büromöbel	Becherstrasse 4, 34343 Oberhuisen
1004	Büroschrank	Aktuell	101	Büromöbel AG	Büroeinrichtungen	Gassengosse 2, 36999 Schwerten

Aufgabe 9

Ein Betrieb mit Außendienstmitarbeitern erfasst ihre Reisekosten in einer Tabelle, die wie folgt aufgebaut ist.

Reisekosten									
Rechnungs-nummer	Datum	Name	Vorname	Straße	PLZ	Ort	Kostenart	Anzahl	Einzelvergütung

Normalisieren Sie sie und dokumentieren Sie ausführlich Ihre jeweiligen Schritte mit Verweis auf die Fachbegriffe.

Überlegen Sie sich am Ende ferner, welche Verbindungen zwischen welchen Tabellen mit entsprechender Kardinalität auftreten könnten.

Aufgabe 10

Eine Tabelle sei in Kurzschreibweise wie folgt gegeben:

RECHNUNG(RechNr, KundenNr, Datum, Artikel)

Denken Sie sich exemplarische Datenbestände aus, die der 1. Normalform nicht genügen. Lösen Sie diese dann auf, bilden Sie also die 1. NF der Tabelle.



Lösungsheft

Kapitel 3
–
Das relationale Modell

Aufgaben

Aufgaben zu Operatoren

Aufgabe 1

Gegeben seien drei Relationen mit den folgenden Tupeln:

Besucht

Gast	Bistro
Hans	Uno
Ede	Uno
Ede	Dos
Ede	Chico
Karl	Dos
Karl	Chico
Heini	Uno

Serviert

Bistro	Getränk
Uno	Wasser
Uno	Kaffee
Dos	Kaffee

Mag

Gast	Getränk
Hans	Wasser
Ede	Wasser
Ede	Kaffee
Karl	Kaffee

a) Bilden Sie Serviert × Mag.

b) Bilden Sie *Serviert* $\rhd\lhd$ *Mag.* Welche Informationen beinhaltet diese Relation?

c) Geben Sie alle Bistros aus, die Getränk servieren, die Karl mag. Überprüfen Sie Ihre Operation in der Relationenalgebra anhand des Beispiels.

d) Geben Sie alle Gäste aus, die mindestens ein Bistro besuchen, die auch das Getränk serviert, das sie mögen. Formulieren Sie die Anfrage mit Operationen der Relationenalgebra.

Aufgabe 2

Gegeben seine folgende Relationen (# ist das Zeichen für *Nummer*):

Lieferanten (L#, LName, Status, Stadt)

Teile (T#, TName, Farbe, Gewicht, Stadt)

Projekte (P#, PName, Stadt)

Lieferungen (L#, T#, P#, Anzahl)

Hierbei bedeutet Stadt einmal die Stadt, in der ein Lieferant sitzt, die Stadt, in der das entsprechende Teil hergestellt wird, bzw. die Stadt, in der ein Projekt stattfindet. Lösen Sie die folgenden Aufgaben durch Operationen aus der Relationenalgebra:

a) Finde Sie alle Lieferungen mit Anzahlen zwischen 300 und 750 und geben Sie alle dazu in der Relation Lieferungen verzeichneten Informationen aus.

b) Geben Sie alle Städte aus, in denen Lieferanten sitzen.

c) Geben Sie alle vorkommenden Paarungen TName, Stadt aus.

d) Finden Sie alle schwarzen Teile. Geben Sie ihre Nummer und ihren Namen aus.

e) Finden Sie alle Lieferanten, die in einer Einzellieferung mehr als 150 Teile geliefert haben. Geben Sie ihren Namen aus.

f) Finden Sie alle Teile, die von Lieferanten in London geliefert wurden. Geben Sie davon die Teilenummer (Teilenamen)aus.

g) Finden Sie alle Orte, in denen sowohl Projekte als auch Lieferanten beheimatet sind.

h) Finden Sie alle Projekte, die mindestens einen Lieferanten für das Projekt im gleichen Ort haben. Geben Sie die Projektnummer aus.

i) Finden Sie alle Teile, die der Lieferant Lux geliefert hat. Geben Sie alle Teilinformationen von diesen Teilen aus.

Aufgabe 3

Gegeben sind die folgenden Tabellen:

Tabelle1

A	B	C
4	2	8
2	2	1
6	7	3

Tabelle2

C	D
8	2
3	6

Tabelle3

B	E
5	3
4	4
5	4

Tabelle4

B	C	D
1	2	1
4	2	1
1	2	9

Führen Sie folgende relationalen Operationen durch und stellen Sie die Ergebnistabelle auf! Beschreiben Sie die Aufgabenstellung mittels der behandelten Symbolik!

a) Selektion von Tabelle1 mit der Bedingung B=2
b) Projektion von Tabelle3 auf E
c) Join Tabelle1 und Tabelle2 nach dem gemeinsamen Attribut C
d) (Selektion von Tabelle 1 mit B>C) vereinigt mit (Selektion von Tabelle1 mit A<5)

Aufgabe 4

Gegeben seien folgende Tabellen

GK-Fach 1

Raum	Fach	Lehrer
137	Mathematik	Müller
221	Deutsch	Schmidt
104	Englisch	Lehmann

GK-Fach 2

Raum	Fach	Lehrer
127	Informatik	Müller
104	Englisch	Lehmann
123	Physik	Paulsen
018	Musik	Schmidt

Themen

Nr.	Thema	Klasse
001	Analysis	12/I
002	Klassik	13/1

Bilden Sie

a) GK-Fach 1 $\cap$ GK-Fach 2

b) GK-Fach 1 $\cup$ GK-Fach 2

c) GK-Fach 1 \ GK-Fach 2

d) Themen x GK-Fach 2

e) σ Name = 'Müller'(Schüler)

f) π Name(Schüler)

g) π Vorname(σ Name = 'Müller'(Schüler))

h) Join(Schüler, Kurs)

Lösungen

Aufgabe 1

a) Bilden Sie Serviert × Mag.

Bistro	Getränk	Gast	Getränk
Uno	Wasser	Hans	Wasser
Uno	Wasser	Ede	Wasser
Uno	Wasser	Ede	Kaffee
Uno	Wasser	Karl	Kaffee
Uno	Kaffee	Hans	Wasser
Uno	Kaffee	Ede	Wasser
Uno	Kaffee	Ede	Kaffee
Uno	Kaffee	Karl	Kaffee
Dos	Kaffee	Hans	Wasser
Dos	Kaffee	Ede	Wasser
Dos	Kaffee	Ede	Kaffee
Dos	Kaffee	Karl	Kaffee

b) weiß unterlegte Tupel

c) $\pi_{Bistro}(\sigma_{Gast = Karl}(Serviert \bowtie Mag)$ oder effizienter

$\pi_{Bistro}(Serviert \bowtie \pi_{Getränk}(\sigma_{Gast = Karl}(Mag)))$

1. Tabellen verbinden $\bowtie$
2. auf Karl reduzieren σ
3. Frage beantworten π

d) $\pi_{Gast}(Besucht \bowtie (\pi_{Bistro, Gast}(Serviert \bowtie Mag)$

Aufgabe 2

In vielen Fällen wurde versucht, die Größe der Zwischentabellen durch frühzeitige Projektion oder Selektion zu minimieren.

a) $\sigma_{Anzahl \geq 300 \text{ und } Anzahl \leq 750}(Lieferungen)$

b) $\pi_{Stadt}(Lieferanten)$

c) $\pi_{TName, Stadt}(Teile)$

d) $\pi_{T\#, TName}(\sigma_{Farbe = schwarz}(Teile))$

e) $\pi_{Lname}(\pi_{L\#}(\sigma_{Anzahl > 150}(Lieferungen)) \bowtie \pi_{L\#, Lname}(Lieferanten))$

f) $\pi_{T\#}(\pi_{L\#}(\sigma_{Stadt = London}(Lieferanten)) \bowtie \pi_{L\#, T\#}(Lieferungen))$

$\bowtie$ $\bowtie$

$\pi_{TName} (\pi_{L\#} (\sigma_{Stadt = London} (\text{Lieferanten})) \quad \pi_{L\#, T\#} (\text{Lieferungen}) \quad \text{Teile})$

g) $\pi_{Stadt} (\text{Lieferanten} \bowtie \text{Projekte})$

h) $\pi_{P\#} (\pi_{P\#, L\#} (\text{Lieferungen}) \bowtie \pi_{P\#, Stadt} (\text{Projekte}) \bowtie \pi_{L\#, Stadt} (\text{Lieferanten}))$

i) $\pi_{T\#} (\pi_{L\#} (\sigma_{Lname = Lux} (\text{Lieferanten}) \bowtie \pi_{L\#, P\#} (\text{Lieferungen})) \bowtie \text{Teile}$

Aufgabe 3

a)

A	B	C
4	2	8
2	2	1

b)

E
3
4

c)

A	B	C	D
4	2	8	2
6	7	3	6

d)

A	B	C
4	2	8
2	2	1
6	7	3

Aufgabe 4

Gegeben seien folgende Tabellen

GK-Fach 1

Raum	Fach	Lehrer
137	Mathematik	Müller
221	Deutsch	Schmidt
104	Englisch	Lehmann

GK-Fach 2

Raum	Fach	Lehrer
127	Informatik	Müller
104	Englisch	Lehmann
123	Physik	Paulsen
018	Musik	Schmidt

Bilden Sie

a) GK-Fach 1 $\cap$ GK-Fach 2

Der Durchschnitt A $\cap$ B zweier Tabellen A und B ist die Menge aller Tupel, die sowohl in A als auch in B enthalten sind.

Raum	Fach	Lehrer
104	Englisch	Lehmann

b) GK-Fach 1 $\cup$ GK-Fach 2

Die Vereinigung A $\cup$ B zweier Tabellen A und B ist die Menge aller Tupel, die in A oder in B oder in beiden Relationen enthalten sind.

GK-Fach 1

Raum	Fach	Lehrer
137	Mathematik	Müller
221	Deutsch	Schmidt
104	Englisch	Lehmann

GK-Fach 2

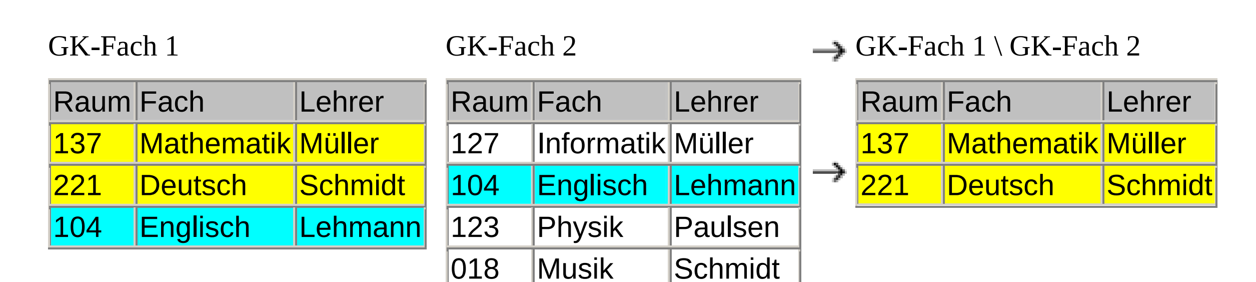
Raum	Fach	Lehrer
127	Informatik	Müller
104	Englisch	Lehmann
123	Physik	Paulsen
018	Musik	Schmidt

→ GK-Fach 1 ∪ GK-Fach 2

Raum	Fach	Lehrer
137	Mathematik	Müller
221	Deutsch	Schmidt
104	Englisch	Lehmann
127	Informatik	Müller
123	Physik	Paulsen
018	Musik	Schmidt

c) GK-Fach 1 \ GK-Fach 2

Die Differenz A \ B zweier Tabellen A und B ist die Menge aller Tupel, die in A aber nicht gleichzeitig in B enthalten sind.



GK-Fach 1

Raum	Fach	Lehrer
137	Mathematik	Müller
221	Deutsch	Schmidt
104	Englisch	Lehmann

GK-Fach 2

Raum	Fach	Lehrer
127	Informatik	Müller
104	Englisch	Lehmann
123	Physik	Paulsen
018	Musik	Schmidt

→ GK-Fach 1 \ GK-Fach 2

Raum	Fach	Lehrer
137	Mathematik	Müller
221	Deutsch	Schmidt

d) Themen x GK-Fach 2

Das Produkt A x B zweier Tabellen A vom Grad n und B vom Grad m ist die Menge aller Kombinationstupel (n x m-Tupel).

Themen

Nr.	Thema	Klasse
001	Analysis	12/I
002	Klassik	13/1

GK-Fach 2

Raum	Fach	Lehrer
127	Informatik	Müller
104	Englisch	Lehmann
123	Physik	Paulsen
018	Musik	Schmidt

→

Nr.	Thema	Klasse	Raum	Fach	Lehrer
001	Analysis	12/I	127	Informatik	Müller
001	Analysis	12/I	104	Englisch	Lehmann
001	Analysis	12/I	123	Physik	Paulsen
001	Analysis	12/I	018	Musik	Schmidt
002	Klassik	13/1	127	Informatik	Müller
002	Klassik	13/1	104	Englisch	Lehmann
002	Klassik	13/1	123	Physik	Paulsen
002	Klassik	13/1	018	Musik	Schmidt

e) σ Name = 'Müller'(Schüler)

Bei der Selektion werden Zeilen aus einer Tabelle ausgewählt, die bestimmten Eigenschaften genügen.

Schüler

SNr	Vorname	Name
4711	Paul	Müller
0815	Erich	Schmidt
7472	Sven	Lehmann
1234	Olaf	Müller
2313	Jürgen	Paulsen

→

σ Name = 'Müller'(Schüler)

SNr	Vorname	Name
4711	Paul	Müller
1234	Olaf	Müller

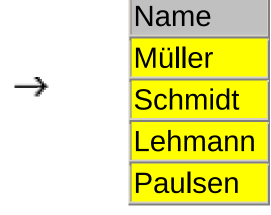
f) π Name(Schüler)

Bei der Projektion werden Spalten aus einer Tabelle ausgewählt, die bestimmten Eigenschaften genügen. Ist ein Eintrag mehrfach vorhanden, so wird er nur einmal angezeigt.

π Name(Schüler)

Schüler

SNr	Vorname	Name
4711	Paul	Müller
0815	Erich	Schmidt
7472	Sven	Lehmann
1234	Olaf	Müller
2313	Jürgen	Paulsen



→

Name
Müller
Schmidt
Lehmann
Paulsen

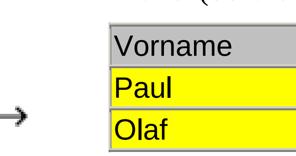
g) π Vorname(σ Name = 'Müller'(Schüler))

Aus der Tabelle Schüler sollen die Vornamen aller Schüler angezeigt werden, deren Nachname Müller ist. Die Abfrage hat also die Form: π Vorname(σ Name = 'Müller'(Schüler))

Schüler

π Vorname(σ Name = 'Müller'(Schüler))

SNr	Vorname	Name
4711	Paul	Müller
0815	Erich	Schmidt
7472	Sven	Lehmann
1234	Olaf	Müller
2313	Jürgen	Paulsen



→

Vorname
Paul
Olaf

h) Join(Schüler, Kurs

Ein Join ist das Verbinden von zwei Relationen zu einer neuen Tabelle.

Die Literatur unterscheidet eine größere Anzahl Joins, für die Schule sind nur die mit

Schüler

SNr	Vorname	Name
4711	Paul	Müller
0815	Erich	Schmidt
7472	Sven	Lehmann
1234	Olaf	Müller
2313	Jürgen	Paulsen

Kurse

SNr	KNr	Fehlstunden	Punkte
0815	03	0	12
4711	03	12	03
1234	23	3	14
0987	09	9	09

→

$Join_{SNr}$(Schüler, Kurs)

SNr	Vorname	Name	KNr	Fehlstunden	Punkte
0815	Erich	Schmidt	03	0	12
4711	Paul	Müller	03	12	03
1234	Olaf	Müller	23	3	14

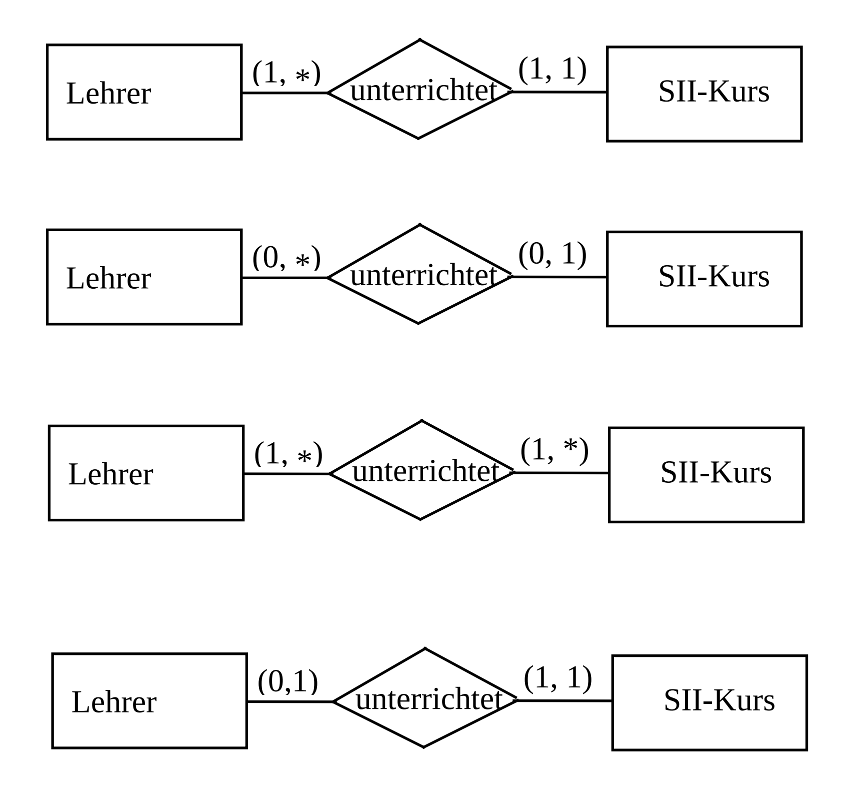
Da zu den SNr.7472, 2314 und 0987 der Tabelle Schüler bzw. Kurse keine "Gegenstücke" in der anderen Tabelle existieren, werden diese in der neuen Relation nicht angezeigt.

Aufgaben

Aufgaben zur Umsetzung in das relationale Modell

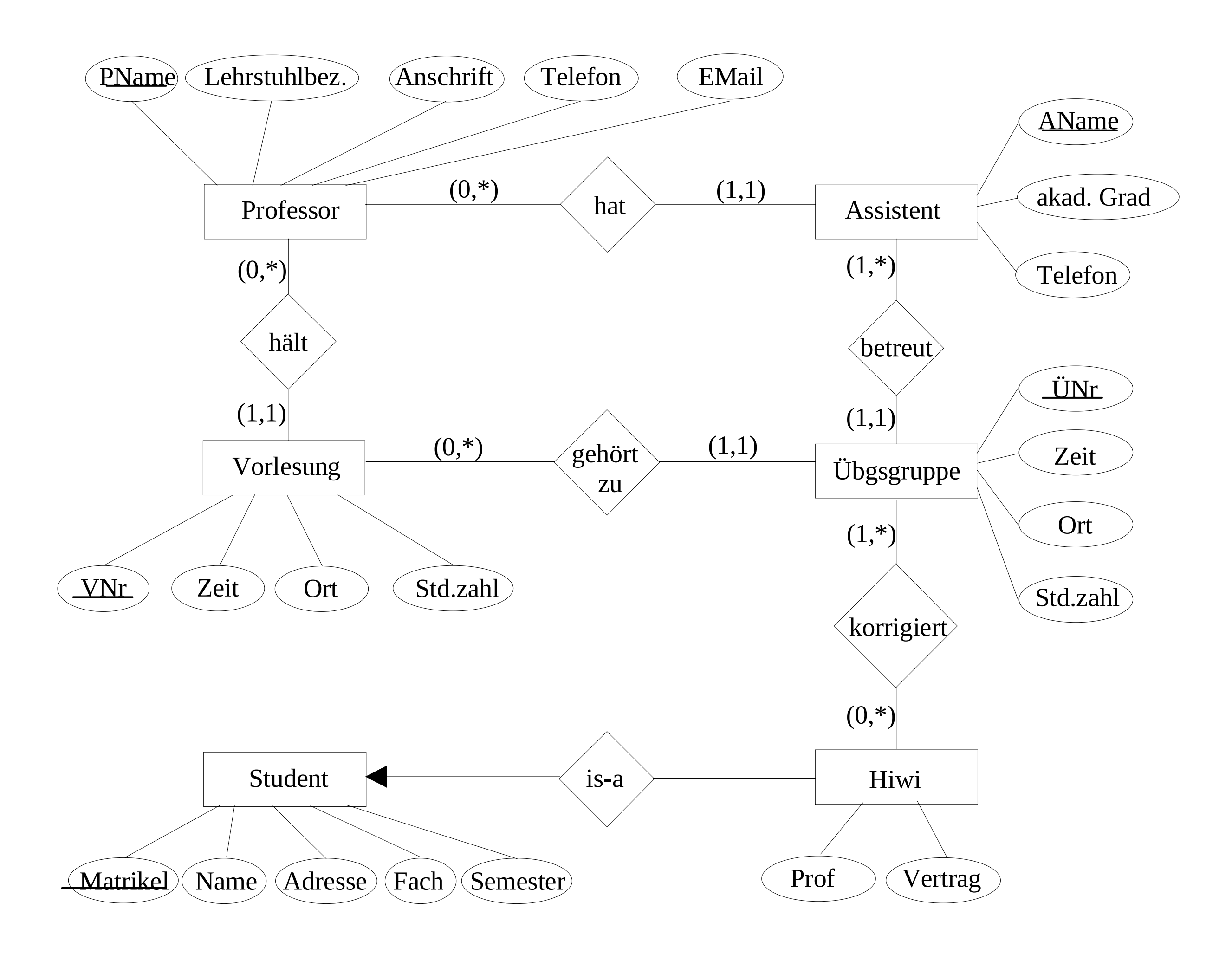
Aufgabe 1

Was bedeuten die folgenden vier Diagramme (Prosa-Erklärung)? Die Attribute wurden zur Vereinfachung weggelassen. Geben Sie jeweils die Komplexität in der Form 1:1, 1:n bzw. n:m an.



Aufgabe 2

Bilden Sie dieses ER-Diagramm in das relationale Modell ab.



Aufgabe 3

Setzen Sie das ER-Diagramm Schule vollständig in ein relationales Modell um, indem Sie alle benötigten Tabellen mit ihren Köpfen angeben, die Primärschlüssel kennzeichnen und durch Pfeile zwischen den Attributen verschiedener Tabellen die Beziehungen verdeutlichen. Im Text wurden schon einige Tabellen eingeführt.

Aufgabe 4

Setzen Sie die unter “Modellierung von Ausschnitten der Wirklichkeit“ dargestellte Aufgabe der Firma Microtec in das relationale Modell um, indem Sie das ER-Modell nach den beschriebenen Verfahren in Tabellen umsetzen!

Aufgabe 5

Setzen Sie das ER-Modell der Spedition Hugo Unbedarft aus „Aufgaben zum ERM“ in ein relationales Modell um. Überlegen Sie dabei, welche Tabellen entstehen müssen und welche Beziehungen sich integrieren lassen.

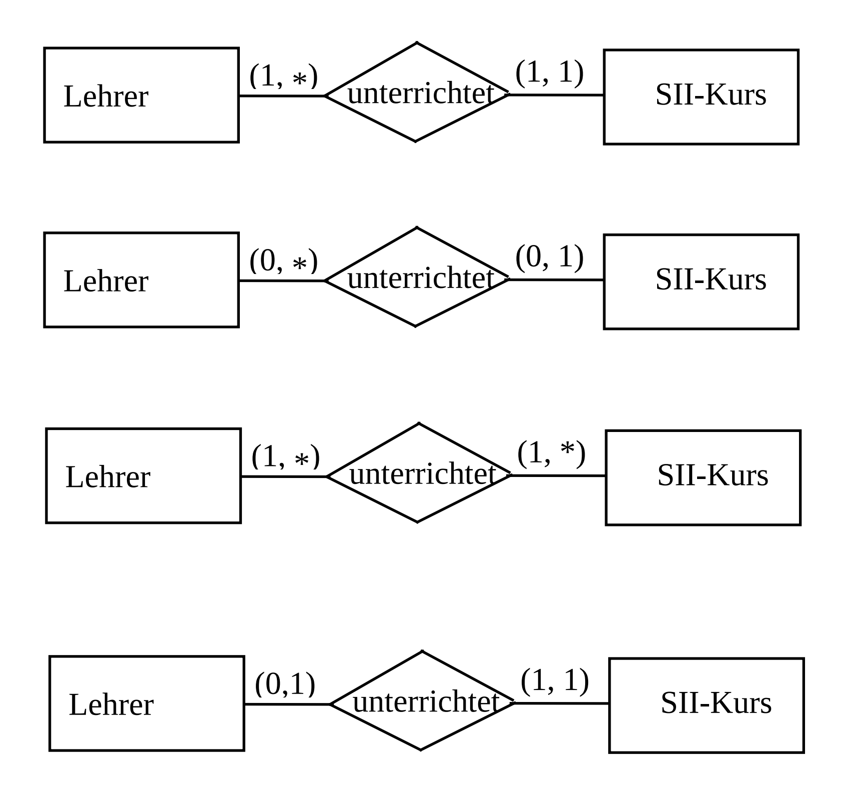
Anmerkung:
Wenn Sie die Normalisierung erlernt haben kommen sie auf diese Aufgabe zurück und Überprüfen anschließend, inwieweit sich die entstandenen Tabellen in der 3. Normalform befinden! Treten Abhängigkeiten zwischen den Schlüsseln auf?

Lösungen

Aufgaben zur Umsetzung in das relationale Modell

Aufgabe 1

Was bedeuten die folgenden vier Diagramme (Prosa-Erklärung)? Die Attribute wurden zur Vereinfachung weggelassen. Geben Sie jeweils die Komplexität in der Form 1:1, 1:n bzw. n:m an.



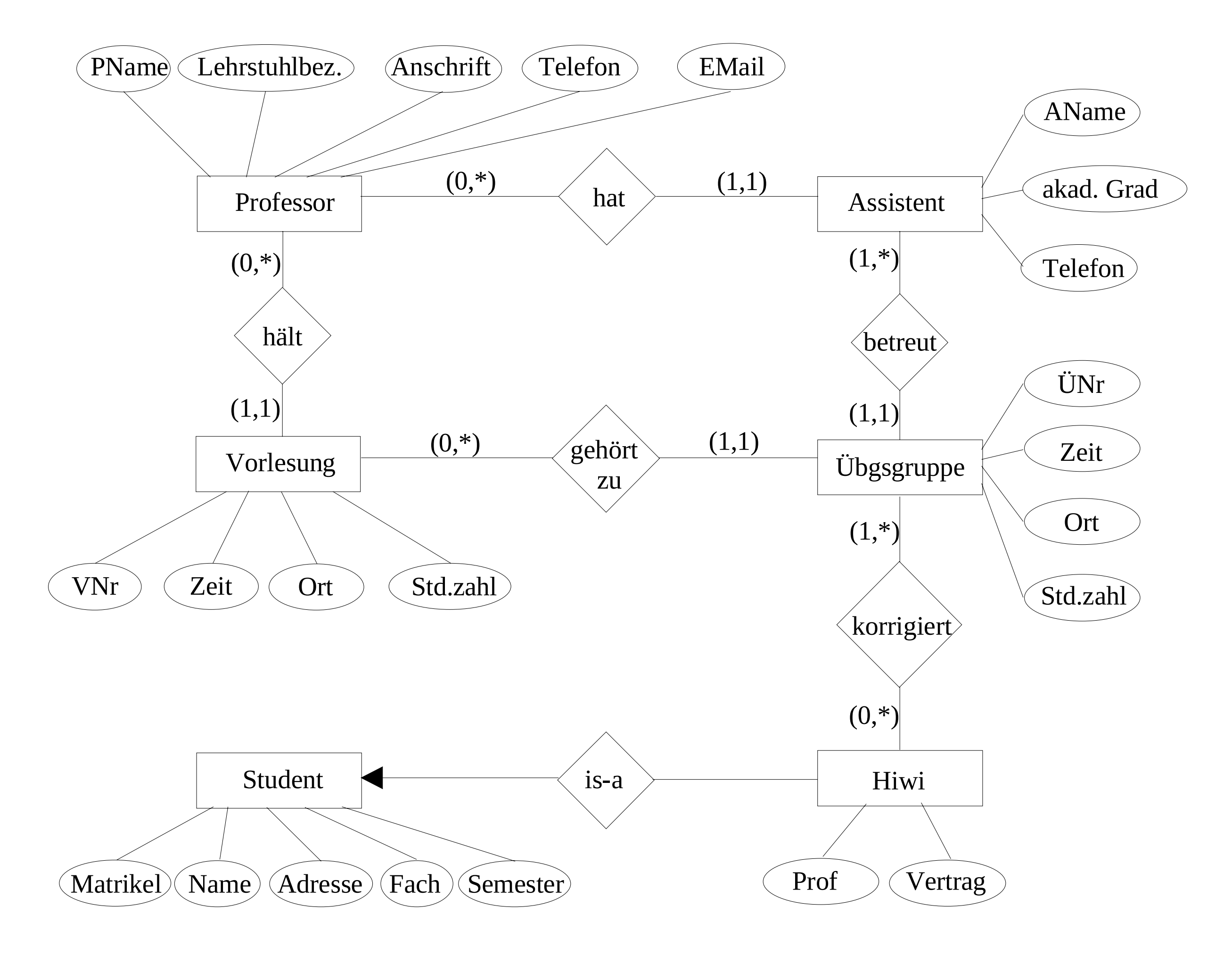
a) Jeder Lehrer hat mindestens einen Kurs, und jeder Kurs hat genau einen Lehrer. 1:n
b) Es gibt auch Kurse ohne Lehrer und Lehrer ohne Kurs. Ein Kurs hat höchstens einen Lehrer. 1:n
c) Jeder Lehrer hat mindestens einen Kurs, und ein Kurs kann mehrere Lehrer haben. n:m
d) Jeder Lehrer hat höchstens einen Kurs, und ein Kurs hat genau einen Lehrer. 1:1

Aufgabe 2

Bilden Sie dieses ER-Diagramm in das relationale Modell ab.

Professor(PName, Lehrstuhlbez., Anschrift, Telefon, EMail)
Assistent(AName, akad. Grad, Telefon, *PName*)
Vorlesung(VNr, Zeit, Ort, Std.zahl, *PName*)
Übgsgruppe(ÜNr, Zeit, Ort, Std.zahl, *AName*, *VNr*)
Hiwi(*Matrikel* , *PName*, Vertrag,)
ÜbgsgruppeHiwi(*ÜNr*, *Matrikel*)
Student(Matrikel, Name, Adresse, Fach, Semester)

Ausführlich:



I. **Prof—hat—Assi** 1 Prof hat 0..mehrere Assis

1 Assi hat genau 1 Prof als Chef

→ alle Assis sind obligatorisch an der hat-Beziehung beteiligt (1-Seite) und damit mit ihr zusammen zu fügen

also

Prof(<u>PName</u>, …)

Assi(<u>AName</u>, …, <u>PName</u>) !Fremdschlüssel für die Verbindung NICHT VERGESSEN!

II. **Prof—hält—Vorlesung**

Zu jeder Vorlesung gehört ein Prof → Vorlesung ist obligatorisch an der Beziehungsrelation 'hält' beteiligt → verschmelzen

also

Prof(unverändert)

Vorlesung(<u>VNr</u>, …, <u>PName</u>) !Fremdschlüssel NICHT VERGESSEN!

III. **Assi—leitet--Übungsgruppen**

Jede Ü-Gruppe wird von genau einem Assi betreut → Ü-Gruppe ist obligatorisch an der Beziehungsrelation 'betreut' beteiligt → verschmelzen

also

Assi(unverändert)

Übungsgruppe(<u>ÜNr</u>, …, AName)

IV. **Übungsgruppen—gehört zu--Vorlesung**

1 Ü-Gruppe gehört zu genau einer Vorlesung → obligatorisch zu 'gehört-zu' → verschmelzen

also

Übungsgruppe(ÜNr, …, AName, VNr)

Vorlesung(unverändert)

V. **Student—IS A--Hiwi**

Ein Hiwi 'ist ein' Student , hat also alle Attribute des Studenten + seine zusätzlichen (speziellen). Hiwi ist der 'speziellere Entitätstyp' → Aufnahme des PS des allgemeinen Entitätstypen in sein Relationsschema

also

Student(Matrikel, …)

Hiwi(Matrikel, PName, Vertrag)   PName = PS des Professors

VI. **Hiwi—korrigiert--Übungsgruppe**

n:m → nicht in weniger als 3 Tabellen erfassbar

also

Hiwi(Matrikel, PName, Vertrag)

Übungsgruppe(ÜNr, …, AName, VNr)

ÜbungsgruppeHiwi(Matrikel, ÜNr)   die Beziehungsrelation 'korrigiert'

Aufgabe 3

Setzen Sie das ER-Diagramm Schule vollständig in ein relationales Modell um, indem Sie alle benötigten Tabellen mit ihren Köpfen angeben, die Primärschlüssel kennzeichnen und durch Pfeile zwischen den Attributen verschiedener Tabellen die Beziehungen verdeutlichen. Im Text wurden schon einige Tabellen eingeführt.

Fremdschlüssel sind kursiv gesetzt

Lehrer (Lehrer-Nr, Name, Vorname, Titel)
Lehrerfach (*Lehrer-Nr*, Fach)
Schüler(Schüler-Nr, Name, Vorname, Geburtsdatum, Schuleintrittsdatum, PLZ, Wohnort, Geschlecht, StraßeNr, *LehrerNr*, Tutorgruppenbezeichnung)
VorherBesSchulen(*Schüler-Nr*, Schule)
Besucht(*Schüler-Nr*, *Kurs-Nr*, Note, Fehlstunden)
Kurs (Kurs-Nr, Thema, Jahrgangsstufe, *Lehrer-Nr*, Fach, Typ)
KursInRaum(*Kurs-Nr*, *Raum-Nr*, Zeit)
Raum(Raum-Nr, Sitzplätze)
Fachraum(Fach, *Raum-Nr*)
FachraumAusstattung(*Raum-Nr*, Sonderausstattung)

Ausführlich:

I. **Lehrer—ist Tutor--Lehrer**

jeder Schüler hat 1 Tutor → obligatorisches Element von 'ist Tutor' → verschmelzen

also

Schüler(<u>Schüler-Nr</u>, Name, Vorname, Geburtsdatum, Schuleintrittsdatum, PLZ, Wohnort, Geschlecht, StraßeNr, *LehrerNr*, Tutorgruppenbezeichnung)

LehrerNr:	Fremdschlüssel
Tutorgruppenbezeichnung	für die Beziehung

Mehrfachattribut 'vorher besuchte Schule' → eigene Tabelle mit PS aus Schüler

aber zusammengesetzter PS nötig, da >1 vorherige Schule pro Schüler

also VorherBesuchteSchule(<u>Schüler-Nr, Schule</u>)

Lehrer(<u>Lehrer-Nr</u>., Titel, Name, Vorname)

Mehrfachattribut 'LehrerFächer' → eigene Tabelle mit PS aus Lehrer

aber zusammengesetzter PS nötig, da >1 Fach pro Lehrer-Nr

also LehrerFächer(<u>Lehrer-Nr</u>., <u>Fach</u>)

II. **Lehrer—besucht--Kurs**

n:m → 3 Tabellen

Schüler(unverändert)

besucht(<u>Schüler-Nr</u>, Kurs-Nr, Note, Fehlstunden)

Schüler-Nr: PS(E1)

Kurs-Nr PS(E2)

Kurs(<u>Kurs-Nr</u>, Thema, Jahrgangsstufe, Fach, Typ)

III. **Lehrer—hält--Kurs**

1 Kurs wird immer von 1 Lehrer gehalten → 'Kurs' ist obligatorisches Element von 'hält'

Lehrer(unverändert)

Kurs(<u>Kurs-Nr</u>., …, <u>Lehrer-Nr</u>)

IV. **Raum—is a--Fachraum**

In das Relationenschema des spezielleren Entity-Typs zusätzlich den PS des allgemeinen Typs aufnehmen, also

Fachraum: spezieller

Raum: allgemeiner → PS von 'Raum' ins Relationenschema von 'Fachraum' aufnehmen

Raum(<u>Raum-Nr</u>., Sitzplätze)

Fachraum(<u>Raum-Nr</u>, <u>Fach</u>)

SonderausstattungenFachraum(<u>Raum-Nr</u>, <u>Sonderausstattung</u>) → Mehrfachattribut

Zusammengesetzter PS, da >1 Sonderausstattung pro Raum

V. **Kurs—in--Raum**

Raum(unverändert)

Raum(unverändert)

KursImRaum(<u>Raum-Nr</u>, <u>Kurs-Nr</u>, <u>Zeit</u>)

1 Kurs in 1 Raum zu verschiedene Zeiten möglich

Aufgabe 4

a) Setzen Sie die unter "Modellierung von Ausschnitten der Wirklichkeit" dargestellte Aufgabe der Firma Microtec in das relationale Modell um, indem Sie das ER-Modell nach den beschriebenen Verfahren in Tabellen umsetzen!

Artikel(<u>Artikel-Nr</u>, Bezeichnung, Verkaufspreis, Bestand)
Liefert(<u>*Artikel-Nr*</u>, <u>*Hersteller-Nr*</u>, Preis)
Hersteller(<u>Hersteller-Nr</u>, Name, Adresse)
Umfaßt(<u>*Bestell-Nr*</u>, <u>*Artikel-Nr*</u>, Stückzahl)
Bestellung(<u>Bestell-Nr</u>, Bestelldatum, *Kunden-Nr*)
Kunde(<u>Kunden-Nr</u>, Name, Adresse)

Aufgabe 5

Setzen Sie das ER-Modell der Spedition Hugo Unbedarft aus „Aufgaben zum ERM" in ein relationales Modell um. Überlegen Sie dabei, welche Tabellen entstehen müssen und welche Beziehungen sich integrieren lassen.

Anmerkung:
Wenn Sie die Normalisierung erlernt haben kommen sie auf diese Aufgabe zurück und Überprüfen anschließend, inwieweit sich die entstandenen Tabellen in der 3. Normalform befinden! Treten Abhängigkeiten zwischen den Schlüsseln auf?

Auftrag(<u>AuftragsNr,</u> Bezeichnung, Ziel, Entfernung)
WirdDurchgeführt(<u>AuftragsNr, LkwNr,</u> Auftragsdatum)
Lkw(<u>LkwNr,</u> Typ, Maxzuladung, TÜV)
WirdGefahren(<u>LkwNr, FahrerNr)</u>
Fahrer(<u>FahrerNr,</u> Name, Vorname, Typ)

Die n:m-Beziehungen werden jeweils in 3 Tabellen abgebildet, diese befinden sich in der 3. NF nach Codd.

Aufgaben

Aufgaben zur Normalisierung

Aufgabe 1

Ein Hochschulberater berät Studenten, die alle im Wohnheim der Universität leben und alle das gleiche Hauptfach studieren. Aus Besprechungen mit Studenten will der Berater eine kleine Datenbank zur Unterstützung der Beratung entwickeln. Er legt folgende Attribute und Regeln fest:

SNr	Studentennummer, ganze Zahl , eindeutiger Schlüssel für Studenten
SName	Name des Studenten, nicht eindeutig
ZNr	Zimmernummer ganzzahlig; jeder Student hat ein Zimmer, das aber von mehreren Studenten bewohnt werden kann
TNr	Telefonnummer des Studenten; sie ist für alle Zimmerbewohner gleich
Kurs	Identifikationsnummer der Kurse, die ein Student belegt oder bereits abgeschlossen hat
Semest	Semester, in dem ein Kurs abgeschlossen wurde; ein Student darf den gleichen Kurs in einem späteren Semester wiederholen
Note	Note eines abgeschlossenen Kurses

Der Berater möchte folgende Daten in der Datenbank speichern (Auszug):

SNr	SName	ZNr	TNr	Kurs	Semest	Note
3215	Jonas Mike	120DH	2136	MAT122	W88	1,4
				PHY120	S88	2,5
				WIW330	W89	3,1
				MAT122	S89	1,2
3456	Schmidt Klaus	237VH	3127	MAT122	W87	3,2
				MAT130	S87	2,9
4576	Neider Paul	120DH	2136	PHY230	W88	2,8
				MAT120	S88	2,1

a) Erstellen Sie ein Diagramm zum ER-Modell für diese Daten.

b) Machen Sie aus der obigen Tabelle eine gültige Relation in der 1. Normalform. Welche Anomalien können dabei auftreten?

c) Erstellen Sie anhand des ER-Modells und der Umsetzungsregeln das relationale Modell. Begründen Sie, inwieweit die Tabellen die 3. Normalform erfüllen.

d) Geben Sie die relationalen Operationen an, um Kurslisten auszugeben.

Aufgabe 2

Ein kleines Belegkrankenhaus stellt auf EDV um. Die Patientendaten enthalten die folgenden Attribute in Tupelschreibweise:

PATIENT = (Name,Gebdatum, Adresse, Blutdruck, Datum, Diagnose, Therapie, Datum, Diagnose, Therapie,...,Belegung)

Anmerkung: *Systol*: zusammenziehen; Kontraktion eines Hohlorgans

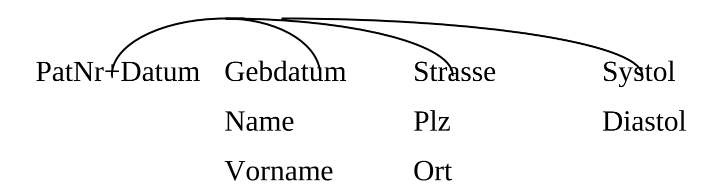
Diastol: erschlaffen; Erschlaffen eines muskulären Organs zwischen 2 Systolen

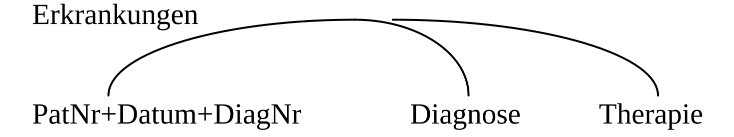
Beides bezieht sich auf den Blutdruck.

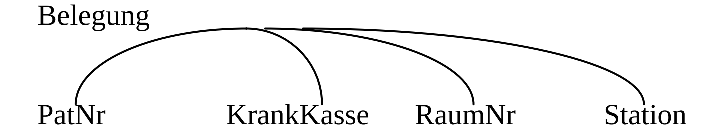
Erklären Sie anhand der folgenden Abbildungen den Gang der Normalisierung und begründen Sie die einzelnen Schritte. Bilden Sie dabei Klassen , gliedern Sie also die Attribute unter Oberbegriffen.

1. Normalform

Patient

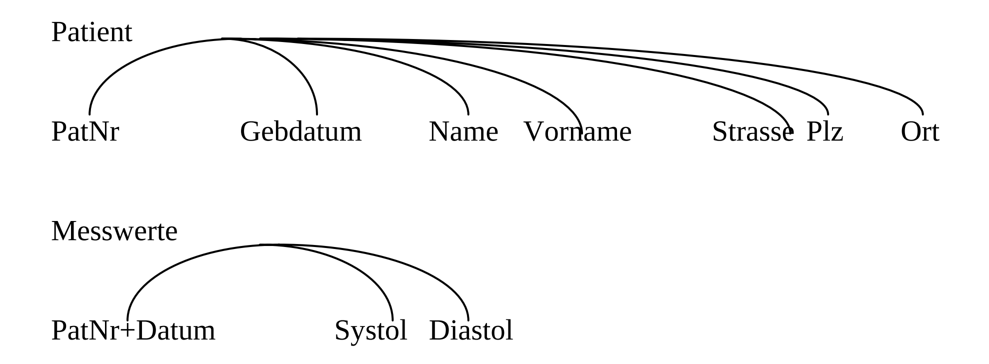






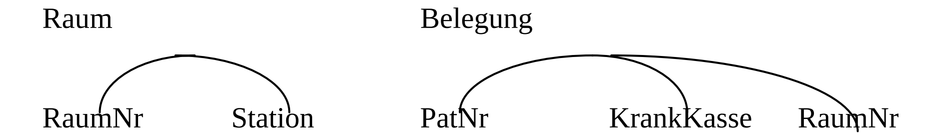
2. Normalform

Die Datengruppe PATIENT wird weiter zerlegt:



3. Normalform

Die Datengruppe BELEGUNG wird weiter zerlegt:



Geben Sie für die 3. Normalform die entsprechenden Tabellen im relationalen Modell an!

Aufgabe 3

Sie haben die Aufgabe der Spedition von Hugo Unbedarf bereits gelöst und sich auch schon über die Tabellen und Verbindungen Gedanken gemacht, bevor Sie dieses Thema explizit erlernt haben. Schauen Sie sich jetzt die Aufgabe noch mal an und lösen Sie sie formal im Lichte Ihres jetzigen Kenntnisstands.

Aufgabe 4

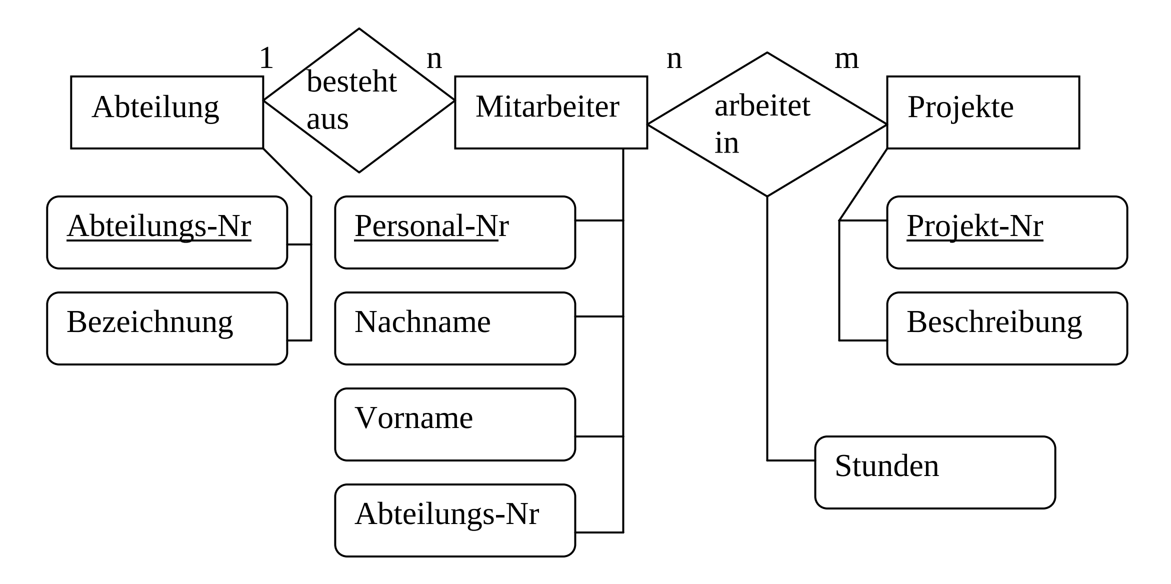
Eine Tabelle mit Lehrerdaten sei wie folgt formuliert

Nr	Nachname	Vorname	Amtsbez.	Besoldungsgruppe	Klasse
1	Bavaria	Eusebia	StRin	A13	11WA, 13ET
2	Bachmann	Hanna	StRin	A13	11WB
3	Kalkulus	Carl-Johann	StD	A15	12SC
4	Spike	Moses	OStR	A14	12GB

Analysieren Sie die Tabelle. Welche Schwächen fallen auf? Normalisieren Sie sie anschließend.

Aufgabe 5

Gegeben sei folgendes ERM.



Geben Sie die sich ergebende relationale Tabellenstruktur in der Kompaktschreibweise “Tabellenname(*PS*, Attribute)“ an. Leiten Sie dabei Ihre Lösung ausführlich her und dokumentieren Sie die einzelnen Schritte.

Aufgabe 6

Sie importieren Daten aus einer Excel-Tabelle und erhalten die folgende Struktur. Normalisieren Sie sie bis zur 3. Normalform. Dokumentieren Sie dabei ausführlich Ihre Schritte mit der entsprechenden Begründung (Definition der Normalformen und ihre Folgerungen).

PersNr	Nachname	Vorname	Abteilung	Projekt	Stunden
1	Lorenz	Christian	Einkauf	Verkaufsanalyse	198, 201
2	Baumann	Peter	Verkauf	Optimierung	120, 189, 43
3	Petersen	Anna	Personal	Weiterbildung	120

Aufgabe 7

Reflektieren Sie nochmals über das systematische Vorgehen während des Normalisierungsprozesses. Dazu studieren Sie vorliegenden Text und ergänzen die sich ergebenden Tabellen.

Anmerkung: Es gibt mehrere denkbare Lösungen, also mehrere Tabellenstrukturen. Falls Ihre Idee nicht in die vorgesehenen Tabellen passt ergänzen Sie einfach die Spaltenanzahl, fragen sich aber kritisch, ob Ihre Lösung den gestellten Anforderungen genügt. Das primäre Ziel der Aufgabe ist die nochmalige intensive Reflexion des systematischen Normalisierungsprozesses.

Ein Bücherwurm Karlo möchte Ordnung in sein ganz privates Bücherchaos bringen, da ihm die Unmengen an über die Jahre erworbenen Büchern mittlerweile ein wenig über den Kopf gewachsen sind. Entsprechend entschließt er sich spontan eine Datenbank (DB) dafür zu

entwickeln. Seine Freundin studiert gerade fleißig Informatik und kommt am Wochenende vorbei, da möchte unser edler Gesell' natürlich ein wahres Kunstwerk mit allen Schikanen vorweisen können.

Als Regeln für gutes Datenbank-Design findet er folgende Weisheiten

- Tabellen sollten keine redundanten (sich wiederholende) Daten enthalten. Wenn in einer Tabelle immer wieder dasselbe eingegeben werden muss, stimmt etwas nicht.
- Bei der Konzeptentwicklung der DB (konzeptuelles Modell) bietet sich z.B. Excel an, da man damit leicht die Tabellen und ihre Attribute visualisieren kann, ohne auf die spezifischen Restriktionen eines echten DBMS achten zu müssen

Diese Regeln sind formalisiert als Normalisierung bekannt.

Als erster Schritt im DB-Design ist der Weltenausschnitt zu modellieren, den man gerade betrachtet. Bei einer Bücherdatenbank sind entsprechend die Attribute zu ermitteln, die in der DB verwaltet werden sollen. Es wären dies beispielsweise

- Titel (title)
- Verlag (publName)
- Veröffentlichungsjahr (year)
- Autoren (authorX)

Bei der Ermittlung dieser Attribute helfen Realweltdaten wie Karteikarten, Akten usw. Eine handschriftliche Bücherliste von Karlo hat beispielsweise folgende Struktur:

title	publName	year	author1	author2	author3
Linux	Addison-Wesley	2004	Meyer		
Web	Apress	2003	Schulz	Einstein	
Mathe	New Riders	2006	Schmitz		
Physik	Einstein-Verlag	1901	Newton	Schrödinger	Heisenberg
Lyrik	Federweiß	2006	Schulz		

Nun ja, also genau genommen hatte bei der Physik auch noch Max Planck seine Finger im Spiel, aber es war halt nur Platz für drei Autoren… Nicht sehr schön. Die Festlegung auf drei Autoren ist willkürlich, unflexibel und schlichtweg unschönes Design. Auch die Lösung alle Autoren in ein einziges Feld authors zu schreiben

authors
Newton, Schrödinger, Heisenberg, Planck

ist nicht so das wahre, da Karlo gehört hat man solle pro Attribut, also pro Spalte in einer Tabelle, nur *atomare Werte*, also ein Datum statt mehrerer, verwalten.
Nun denn, als Designkriterien für diese erste Hürde gelten folgende Weisheiten , die der *ersten Normalform*:

- Nur atomare Werte pro Spalte.
- Spalten mit gleichartigem Inhalt müssen entfernt werden.
- Für jede Gruppe zusammengehöriger Daten muss eine eigene Tabelle gebildet werden.
- Jeder Datensatz muss durch einen sog. Primärschlüssel eindeutig identifiziert werden.

Die dritte Regel ist schon erfüllt, da momentan in jeder Spalte nur Daten gesammelt werden, die auch zu einem Buch gehören. Regel 1 ist gleichfalls erfüllt, da das Problem der Mehrfachdaten pro Attribut von Karlo durch die drei Autorenspalten gelöst wurde, auch wenn sich dadurch Probleme mit Regel zwei eingefangen hat. Regel 4 ist leicht zu befolgen. Es wird einfach eine fortlaufende Nummer eingeführt (AutoWert), die *titleID*, und als Primärschlüssel verwendet. Diese Nummer identifiziert dann das Buch mit all seinen Attributen eindeutig. Entsprechend ergibt sich die Tabelle nun wie folgt:

Prima, alle Kriterien der ersten Normalform sind erfüllt! Selbst das Autorenanzahlproblem wurde gelöst. Allerdings mag es doch verdächtig anmuten alle Daten für n-Autoren n-mal vorrätig zu halten. Das kann nicht der Weisheit letzter Schluss sein.
Eine Tabelle wird in die *zweite Normalform* umgeformt, in dem

- Eine Tabelle in erster Normalform ist.
- Immer dann, wenn sich Inhalte in Spalten wiederholen, müssen die Tabellen in mehrere Teiltabellen zerlegt werden.
- Die Tabellen müssen durch sog. Fremdschlüssel miteinander verbunden werden.

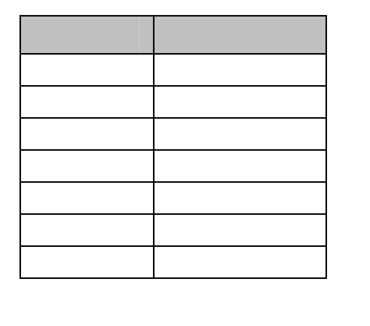
In obiger Tabelle wiederholen sich Daten auf Grund der Autoren. Eine Auslagerung derselbigen ergibt folgende Tabellenstruktur:

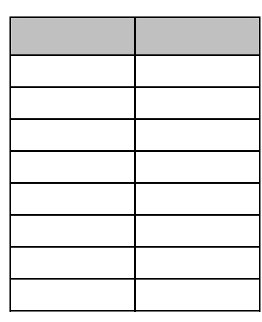
authID	titleID	author
1	1	Meyer
2	2	Schulz
3	3	Schmitz
4	4	Newton
5	4	Schrödinger
6	4	Heisenberg
7	4	Planck
8	5	Schulz

Schon besser. Allerdings kommt 'Schulz' als Autor immer noch mehrmals vor. Dies tritt immer dann auf, wenn ein Autor mehr als ein Buch geschrieben hat. Als Lösung käme eine weitere Aufteilung der Tabelle in Frage:

titles

authors *rel_title_author*





Letztere Tabelle muss von Rechts nach Links gelesen werden. Sie gibt also an, welche Bücher ein Autor – identifiziert über seine eindeutige authID – geschrieben hat. Vom Buch in einer Anfrage geht es also über die titleID zur authID, welche schließlich zu den Autoren des Buches führt. Die Tabelle rel_title_author hat übrigens einen *zusammengesetzten Primärschlüssel*, der aus den beiden Attributen besteht. Er identifiziert jedes Tupel eindeutig, daher ist ein zusätzlicher Primärschlüssel nicht nötig.
Es gibt noch eine weitere wichtige Normalform, die *dritte Normalform*. Diese löst indirekte (*transitive*) Abhängigkeiten auf. Ist A von B abhängig und B von C, so ist A transitiv von C abhängig, was nicht erlaubt ist, wenn eine Tabelle der dritten Normalform genügen soll. Ein typisches Beispiel ist die Postleitzahl und der Ort. In einer Mitarbeitertabelle werden die Personalstammdaten inklusive Wohnort erfasst. Eine MitarbeiterID (A) identifiziert den Mitarbeiter eindeutig, auch die zugehörige PLZ (B). Der Wohnort (C) ist aber abhängig von der PLZ (B) und damit besteht transitive Abhängigkeit zwischen MitarbeiterID und Wohnort. Entsprechend werden PLZ und Ort aus der Tabelle in eine eigene ausgegliedert.

PersID	Name	…	PLZ	Ort
1			33433	Marl
2			33453	Gelsenkirchen
3			2432	New York

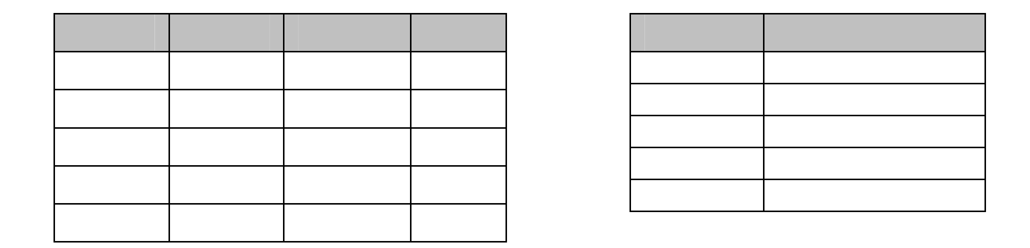
→

PersID	Name	…	PLZ
1			33433
2			33453
3			2432

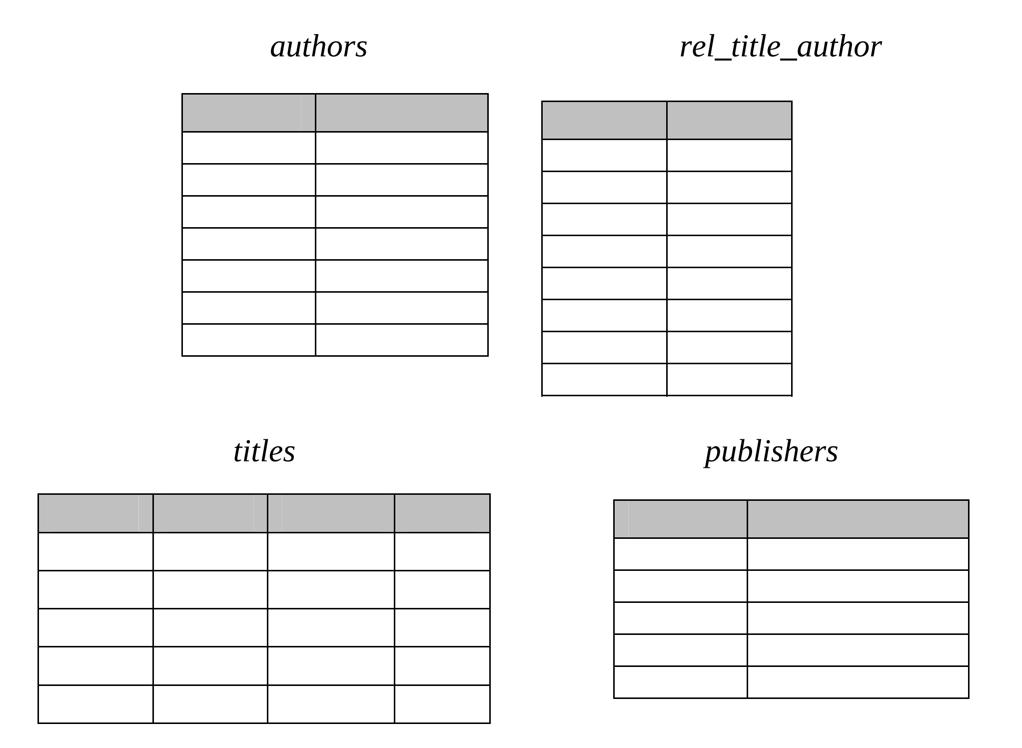
PLZ	Ort
33433	Marl
33453	Gelsenkirchen
2432	New York

Entsprechend reagiert Karlo und findet heraus, dass die Spalte *publName* in der *title*-Tabelle betroffen ist. Die Liste der Verlage und die Liste der Buchtitel sind voneinander abhängig und sollten daher getrennt behandelt werden. Es ist zwar nötig und sinnvoll zu jedem Buch den Verlag zu vermerken, aber es reicht doch ein kurzer Schlüssel statt immer der vollständigen Daten (hohen Fehleranfälligkeit und große Speicherplatzverschwendung, was zudem den Zugriff auf die Daten verlangsamt). Dies ist der Hintergrund für die dritte Normalform.

Gesagt, getan. Karlo macht folgendes daraus.



Die Tabellen *rel_title_author* und *authors* änders sich in der dritten Normalform nicht mehr. Die gesamte Bücherdatenbank besteht nun aus drei Tabellen.



Die Normalformen formalisieren demnach zwei Designprinzipien:

- Fasse in einer Tabelle nur das zusammen, was zusammen gehört.
- Vermeide Redundanzen und Abhängigkeiten durch Separierung in eine eigenen Tabelle.

Erweiterungen:

Die Datenbank könnte noch leicht um einige weitere – realistische – Features erweitert werden. So könnte die *titles*-Tabelle noch weitere Felder erhalten, wie etwa einen *Untertitel* oder einen *Kommentar*. Eine weitere languages-Tabelle könnte eine Liste aller Sprachen enthalten, in denen das Buch erschienen ist. Ein Verweis darauf könnte über ein *langID*-Feld in der Tabelle *titles* erfolgen. Eine Tabelle *categories* könnte eine Liste der Buchkategorien enthalten, die hierarchisch gegliedert ist, z.B. 'computer books → databases' und 'computer books → programming', 'computer books → programming → java' usw. Erfasst werden könnte diese Kategorie-Datailtabelle über ein Feld *catID* in der Tabelle *titles*.

Exkurs: Hierarchien

Categories ordnet Bücher in verschiedene Kategorien, die hierarchisch strukturiert sind. Dabei verweist *parentID* auf die übergeordnete Ebene.

All Books
- Children books
- Computer books
    - Databases
        - Object oriented db
        - Relational db
        - Sql
    - Programming
        - Pearl
        - Php

→

catID	catName	parentID
1	computer books	11
2	databases	1
3	programming	1
…		
11	all books	11

Aufgabe 8

Eine Möbelfirma stellte vom Karteikartensystem

Artikelbezeichnung: **Gabriele**		
Artikelart: **Schreibtisch**	Artikelnummer: **1000**	
Bestand:	Mindestbestand: **5**	Höchstbestand: **10**
Einkaufspreis:	Verkaufspreis:	
Lieferant: **Tranel GmbH**	Name2: **Büromöbel**	
Lieferer_Nr: **102**	Anschrift: **Bechemstr. 67, 47058 Duisburg**	

auf Excel um und möchte nun – mit steigendem Datenaufkommen – eine echte Datenbank nutzen. Es gibt sich nach dem Import folgende Tabellenstruktur. Normalisieren Sie sie.

Artikel_Nr	Artikelart	Artikel_Bez	LiefererNr	Name1	Name2	Anschrift
1000	Schreibtisch	Gabriele	102	Tranel GmbH	Büromöbel	Bechemstr., 33242 Duisburg
1001	Schreibtisch	Modern	101	Büromöbel AG	Büroeinrichtungen	Gutachtstr. 342, 12345 Bremen
1002	Schreibtisch	Exklusiv	102	Tranel GmbH	Büromöbel	Vogtweg 34, 32453 Verlin
1003	Büroschrank	Elegant	100	Wagner GmbH	Büromöbel	Becherstrasse 4, 34343 Oberhuisen
1004	Büroschrank	Aktuell	101	Büromöbel AG	Büroeinrichtungen	Gassengosse 2, 36999 Schwerten

Aufgabe 9

Ein Betrieb mit Außendienstmitarbeitern erfasst ihre Reisekosten in einer Tabelle, die wie folgt aufgebaut ist.

Reisekosten									
<u>Rechnungs-nummer</u>	Datum	Name	Vorname	Straße	PLZ	Ort	<u>Kostenart</u>	Anzahl	Einzelvergütung

Normalisieren Sie sie und dokumentieren Sie ausführlich Ihre jeweiligen Schritte mit Verweis auf die Fachbegriffe.

Überlegen Sie sich am Ende ferner, welche Verbindungen zwischen welchen Tabellen mit entsprechender Kardinalität auftreten könnten.

Aufgabe 10

Eine Tabelle sei in Kurzschreibweise wie folgt gegeben:

RECHNUNG(RechNr, KundenNr, Datum, Artikel)

Denken Sie sich exemplarische Datenbestände aus, die der 1. Normalform nicht genügen. Lösen Sie diese dann auf, bilden Sie also die 1. NF der Tabelle.

Aufgabe 11

Im Folgenden finden Sie nochmals eine kompakte Erklärung des Normalisierungsprozesses. Studieren Sie ihn sorgfältig.

Umsetzung des ERM in das relationale Datenbankmodell

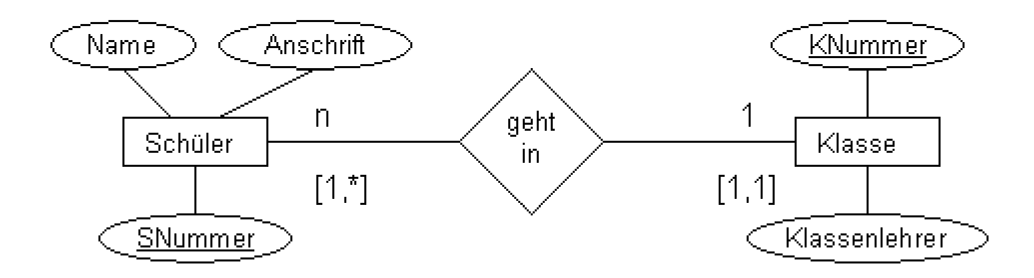
Grundregeln

1. **Entitätstypen:**
   - Jeder Entitätstyp wird in ein eigenes Relationsschema (Tabelle) abgebildet.
   - Schlüssel werden kenntlich gemacht.
2. **Beziehungstypen:**
   - Jeder Beziehungstyp wird in ein eigenes Relationsschema abgebildet.
   - Die Primärschlüssel der beiden beteiligten Entitätstypen werden zu zusätzliche Attributen des Relationsschemas (zus. zu evtl. vorhandenen eigenen Attributen).
   - Der Schlüssel des Relationsschemas bildet sich in Abhängigkeit vom Assoziationstyp wie folgt (Grobklassifikation):

Assoziationstyp	Schlüssel
1:1	einer der Primärschlüssel der beiden beteiligten Entitätstypen
1:n	der Primärschlüssel des zweiten Entitätstyps (also der "n-Entität")
n:m	beide Primärschlüssel der beteiligten Entitätstypen

Somit entstehen aus zwei mit einem Beziehungstyp verbundenen Entitätstypen immer 3 Tabellen.

Beispiel



wird zu

Relationsschema Schüler

SNummer	Name	Anschrift
0815	Schmidt	Mühleweg 7

Relationsschema geht_in

SNummer	KNummer
0815	8a

Relationsschema Klasse

KNummer	Klassenlehrer
8a	Herr Müller

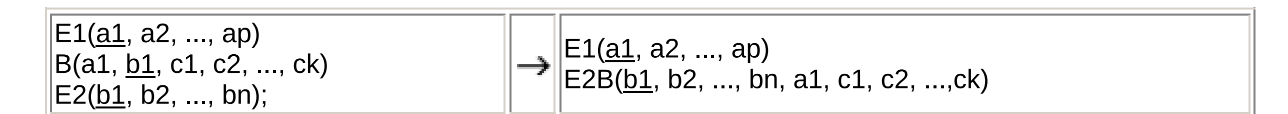
Optimierung

Die entstandenen 3 Tabellen lassen sich optimieren, falls **nicht** der Assoziationstyp n:m vorliegt.

Typ 1:n

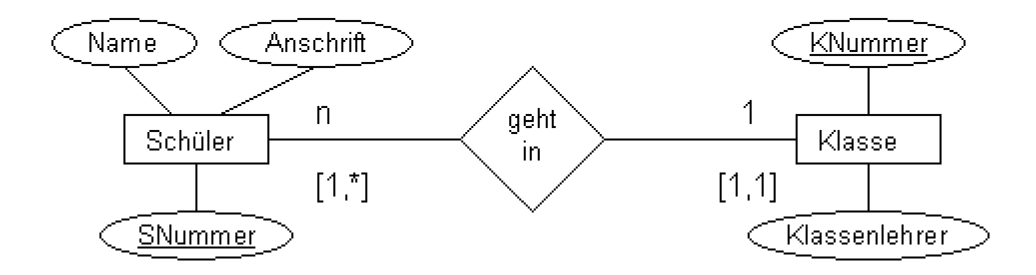
Es lassen sich die drei Relationsschemen u.U. auf zwei reduzieren.

Es gilt dann: Der Beziehungstyp mit all seinen Attribute und der "n-Entitätstyp" bilden eine neue Relation. Schlüssel wird das Schlüsselattribut des Entitätstyps der "n-Seite".



E1(a1, a2, ..., ap) B(a1, b1, c1, c2, ..., ck) E2(b1, b2, ..., bn);	→	E1(a1, a2, ..., ap) E2B(b1, b2, ..., bn, a1, c1, c2, ...,ck)

Beispiel



wird zu

Relationsschema Schüler

SNummer	Name	Anschrift	KNummer
0815	Schmidt	Mühleweg 7	8a

Relationsschema Klasse

KNummer	Klassenlehrer
8a	Herr Müller

Typ 1:c, c:c

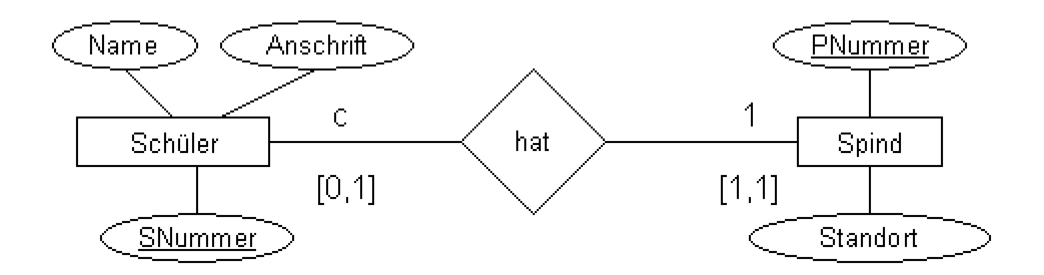
Es lassen sich die drei Relationsschemen auf zwei reduzieren.

Es gilt dann: Der Beziehungstyp mit all seinen Attribute und der "c-Entitätstyp" bilden eine neue Relation. Schlüssel wird das Schlüsselattribut des Entitätstyps der "c-Seite". Bei c:c kann der Fremdschlüssel NULL-Werte enthalten.



E1(a1, a2, ..., an) B(a1, b1, c1, c2, ..., ck) E2(b1, b2, ..., bm);	→	E1(a1, a2, ..., an) E2B(b1, b2, ..., bm, a1, c1, c2, ...,ck)

Beispiel



unter der Voraussetzung, dass jeder Schüler genau ein Spind hat, es aber auch nicht zugeordnete Spinde geben kann, wird zu

Relationsschema Schüler

SNummer	Name	Anschrift	PNummer
0815	Schmidt	Mühleweg 7	4563

Relationsschema Spind

PNummer	Standort
4563	Raum 218

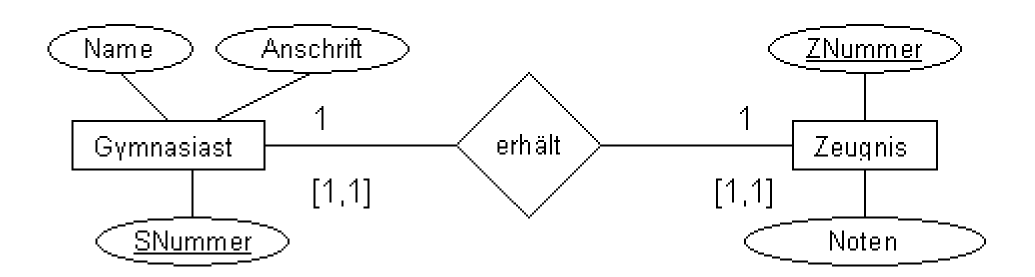
Typ 1:1

Es lassen sich die drei Relationsschemen auf eins reduzieren.

Es gilt dann: Der Beziehungstyp und die beiden Entitätstypen bilden eine neue Relation. Schlüssel wird ein Schlüsselattribut eines Entitätstyps.

E1(a1, a2, ..., an) B(a1, b1, c1, c2, ..., ck) E2(b1, b2, ..., bm);	→	E1BE2(a1, a2, ..., an,b1, b2, ..., bm, c1, c2, ...,ck)

Beispiel



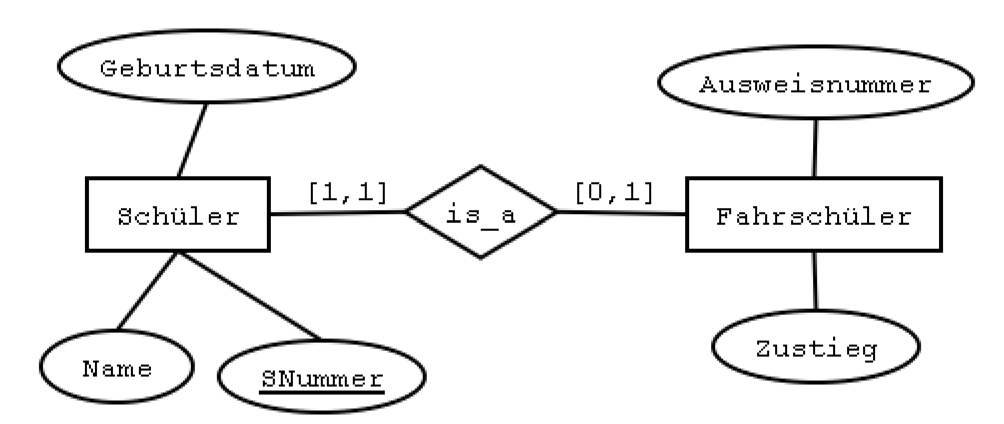
unter der Voraussetzung, dass jeder Gymnasiast auch ein (was auch immer für ein) Zeugnis erhält, wird zu

Relationsschema Gymnasiast_erhaelt_Zeugnis

SNummer	Name	Anschrift	Noten	ZNummer
0815	Schmidt	Mühleweg 7	1,2,...	4711

Sonderfälle

1. **is-a-Beziehungen**
Is-a-Beziehungen drücken Spezialisierungen bzw. Generalisierungen aus. Dabei gilt, dass der Spezialentitätstyp alle Attribute des allgemeinen Entitätstyps erbt. Aus diesem Grund findet sich bei is_a-Beziehungen vom Typ 1:c an einem Entitätstyp (dem Speziellen) keine Schlüsselattributangabe. Is-a-Beziehungen lassen sich auflösen und in eine Tabelle integrieren. Hierzu muss nur der Primärschlüssel des allgemeineren Entitätstyps entweder als Fremdschlüssel (bei 1:mc) oder als Primärschlüssel (bei 1:c) in die Tabelle des speziellen Entitätstyps übernommen werden.



Relationsschema Schüler

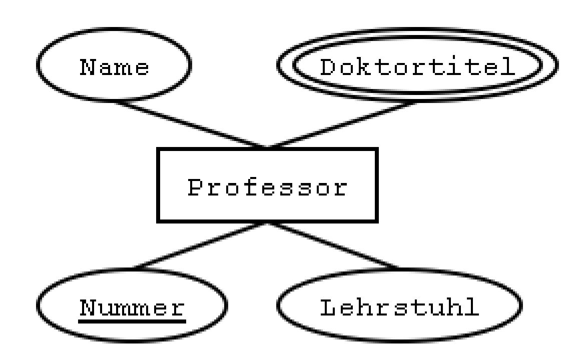
SNummer	Name	Geburtsdatum
0815	Schmidt	09.09.1990
4711	Paulsen	22.07.1987
007	Bond	02.05.1936

Relationsschema Fahrschüler

SNummer	Ausweisnummer	Zustieg
0815	8277728	Bahnhof
007	6525593	Hohenwulsch

2. **Mehrwertige Attribute**

Mehrwertige Attribute werden im relationalen Modell durch eine neue Tabelle dargestellt, die als Attribute das Primärschlüsselattribut des Entitätstyps und das Mehrfachattribut besitzen. Beide Attribute sind Schlüssel der Tabelle.



Relationsschema Professor

Nummer	Name	Lehrstuhl
0815	Schmidt	KI
4711	Paulsen	TI

Relationsschema Doktortitel

Nummer	Titel
0815	Dr. nat.
0815	Dr. phil.
4711	Dr. med. dent.

Lösungen

Aufgaben zur Normalisierung

Aufgabe 1

Die Objekte STUDENT, ZIMMER, KURS stehen über die Beziehungen WOHNT_IN und HAT_BELEGT miteinander in Beziehung. Die Beziehung WOHNT_IN hat die Komplexität 1:n, die Beziehung HAT_BELEGT die Komplexität m:n.

Die Wiederholungsgruppe KURS muß ausgegliedert werden, es entsteht eine neue Tabelle in der 1.NF. Zwischen ZNR und TNR besteht eine direkte Abhängigkeit, welche zu verschiedenen Anomalien führen kann.

Gliedert man die Tabelle mit den Attributen ZNR und TNR aus (1:1 -Beziehung) und integriert SEMESTER in HAT_BELEGT, so erfüllen die Tabellen die 3.NF, da keine Abhängigkeiten mehr vorhanden sind. Die Beziehung WOHNT_IN kann in die Tabelle STUDENT integriert werden, da es eine obligatorische Mitgliedschaft gibt. Da die Kursbezeichnung nicht eindeutig ist, kann entweder mit einem zusammengesetzten Schlüssel gearbeitet werden (KURS, SEMESTER) oder eine eindeutige Kursnummer vergeben werden.

Student(SNr, Sname, ZNr)
ZimmerTelefon(ZNr, TNr)
Kurs(KursNr, Kurs,Semester)
hat_belegt(SNr, KursNr, Note)

Die beteiligten Tabellen STUDENT, KURS und HAT_BELEGT werden über einen Equi-Join verbunden, da SNr bzw. KNr gemeinsame Schlüssel darstellen. Eine Selektion auf die betreffende Kursnummer liefert die Kursliste.

Aufgabe 2

PATIENT = (Name, Gebdatum, Adresse, Blutdruck, Datum, Diagnose, Therapie, Datum, Diagnose, Therapie,...,Belegung)

Die Normalisierung bezieht sich im <u>ersten Schritt</u> auf die Zerlegung des Patiententupels in drei Tupel mit zusammengehörenden Attributen, praktisch den Klassen (aus der OOP) 'Patient, Erkrankungen, Belegung'.

PATIENT =

	PatNr	Name	Therapie1	Blutdruck	Belegung
	Datum	Gebdatum	Datum1		
		Adresse	Diagnose1		
			…		
Klasse	"Name"	"Patientendaten"	"Diagnose"		
Überklassen	Patientendaten, Diagnosedaten				

Nichtatomar: Adresse, Blutdruck, Name, Belegung

Wiederholungsgruppen: Therapie, Datum, Diagnose

→ es gibt nach der 1 NF kein … mehr

- Mehrfachattribute
- nichtatomare Werte

Nichtatomar:

Adresse: Strasse, PLZ, Ort

Blutdruck: Systol, Diastol

Name: Vorname, Nachname

Belegung: PatNr, KrankKasse, RaumNr, Station

Wiederholungsgruppen:

Therapie1 / Datum1 / Diagnose1 …:

Erkrankungen(PatNr, Datum, DiagNr, Diagnose, Therapie)

Im zweiten Schritt wird die Abhängigkeit vom gesamten PS gefordert. Es sind ja durch die 1 NF viele zusammengesetzte PS entstanden. In diesen Tabellen müssen jetzt die Bedingungen der Abhängigkeit vom gesamten PS geprüft werden.

Belegung: PS besteht nur aus 1 Attribut → automatisch in 2 NF

Erkrankungen: funktionale Abhängigkeit vom gesamten PS gegeben

Patient: PatNr identifiziert eindeutig

- GebDatum
- Name
- Vorname
- Strasse
- PLZ
- Ort

während

Systol

Diastol

vom gesamten PS abhängig sind, also

Messwerte PatNr+Datum

- Systol
- Diastol

Im dritten Schritt erfolgt die Auflösung indirekter (transitiver) Abhängigkeiten.

BELEGUNG: → RAUM(RaumNr, Station)

BELEGUNG(PatNr, Krankenkasse, RaumNr)

PATIENT: → PATIENT(PatNr, …, PLZ)

ORTE(PLZ, Ort)

Die Relationen lassen sich direkt aus der Grafik ablesen, da alle Tupel mit ihren Schlüsseln und Attributen bekannt sind und unerwünschte Abhängigkeiten beseitigt wurden.

Patient(PatNr, GebDatum, Name, Vorname, Strasse, Plz)

Orte(Plz, Ort)

Messwerte(PatNr, Datum, Systol, Diastol)

Raum(RaumNr, Station)

Belegung(PatNr, KrankKasse, RaumNr)

Erkrankungen(PatNr, Datum, DiagNr, Diagramme, Therapie)

Aufgabe 3

Sie haben die Aufgabe der Spedition von Hugo Unbedarf bereits gelöst und sich auch schon über die Tabellen und Verbindungen Gedanken gemacht, bevor Sie dieses Thema explizit erlernt haben. Schauen Sie sich jetzt die Aufgabe noch mal an und lösen Sie sie formal im Lichte Ihres jetzigen Kenntnisstands.

Auftrag(AuftragsNr, Bezeichnung, Ziel, Entfernung)
WirdDurchgeführt(AuftragsNr, LkwNr, Auftragsdatum)
Lkw(LkwNr, Typ, Maxzuladung, TÜV)
WirdGefahren(LkwNr, FahrerNr)
Fahrer(FahrerNr, Name, Vorname, Typ)

Die n:m-Beziehungen werden jeweils in 3 Tabellen abgebildet, diese befinden sich in der 3. NF nach Codd.

Aufgabe 4

Eine Tabelle mit Lehrerdaten sei wie folgt formuliert

Nr	**Nachname**	**Vorname**	**Amtsbez.**	**Besoldungsgruppe**	**Klasse**
1	Bavaria	Eusebia	StRin	A13	11WA, 13ET
2	Bachmann	Hanna	StRin	A13	11WB
3	Kalkulus	Carl-Johann	StD	A15	12SC
4	Spike	Moses	OStR	A14	12GB

Analysieren Sie die Tabelle. Welche Schwächen fallen auf? Normalisieren Sie sie anschließend.

Redundanzen: Ax taucht mehrmals auf (hier: A13)

→ Zusammenhang Amtsbez. ←→ Besoldungsruppen einmal in eigener Tabelle erfassen

Einfügeanomalie: Es kann keine neue Ax ohne einen Lehrer (Nr) geben, der sie hat (Situation z.B. bei DB-Neuanlage).

Löschanomalie: Ohne Mitarbeiter (alle gelöscht) sind auch die Amtsbez. und die Besoldungsgruppen weg.

Änderungsanomalie: Wird Ax geändert, so muss jeder einzelne Datensatz kontrolliert werden statt es einmal zentral zu formulieren durch Zuordnung Amtsbez. ←→ Besoldungsruppen

Transitive Abhängigkeit: Besoldungsgruppe ist nur von Amtsbez. Abhängig, nicht vom PS.

1. NF: Atomare Werte

nichtatomare Werte gibt es in der 1.Zeile

<u>**Nr**</u>	**Nachname**	**Vorname**	**Amtsbez.**	**Besoldungsgruppe**	<u>**Klasse**</u>
1	*Bavaria*	*Eusebia*	*StRin*	*A13*	*11WA*
1	*Bavaria*	*Eusebia*	*StRin*	*A13*	*13ET*
2	Bachmann	Hanna	StRin	A13	11WB
3	Kalkulus	Carl-Johann	StD	A15	12SC
4	Spike	Moses	OStR	A14	12GB

!!! Die Nr identifiziert jetzt NICHT mehr EINDEUTIG einen Datensatz (Zeile, Tupel). Daher ist der neue PS ein zusammengesetzter PS aus den Attributen *Nr* und *Klasse*. !!!

Sollte ein Lehrer in einer Klasse mehr als ein Fach unterrichten, so wäre der PS um das Attribut *Fach* zu ergänzen.

2. NF: Jedes Nicht-Schlüssel-Attribut ist funktional abhängig vom Gesamtschlüssel, nicht dagegen von Schlüsselteilen

Der PS ist zusammengesetzt, daher ist die 2.NF überhaupt noch zu prüfen.

Nachname, Vorname, Amtsbez. Und Besoldungsgruppe haben aber nichts mit den Klassen zu tun, die hängen nur von der Nr. ab. Ausgliedern in eine eigene Tabelle mit PS als Fremdschlüssel.

<u>**Nr**</u>	**Nachname**	**Vorname**	**Amtsbez.**	**Besoldungsgruppe**
1	Bavaria	Eusebia	StRin	A13
1	Bavaria	Eusebia	StRin	A13
2	Bachmann	Hanna	StRin	A13
3	Kalkulus	Carl-Johann	StD	A15
4	Spike	Moses	OStR	A14

<u>**Nr**</u>	<u>**Klasse**</u>
1	11WA
1	13ET
3	11WB
4	12SC
5	12GB

3.NF: Es gibt keine funktionalen Abhängigkeiten zwischen Nicht-Schlüssel-Attributen, es gibt also keine transitiven Abhängigkeiten.

<u>Nr</u>	**Nachname**	**Vorname**	**Amtsbez.**	**Besoldungsgruppe**

<u>Nr</u>	**<u>Klasse</u>**

Es gibt also transitive Abhängigkeiten. Auflösen durch eigene Tabelle

<u>Nr</u>	**Nachname**	**Vorname**	**Amtsbez.**
1	Bavaria	Eusebia	StRin

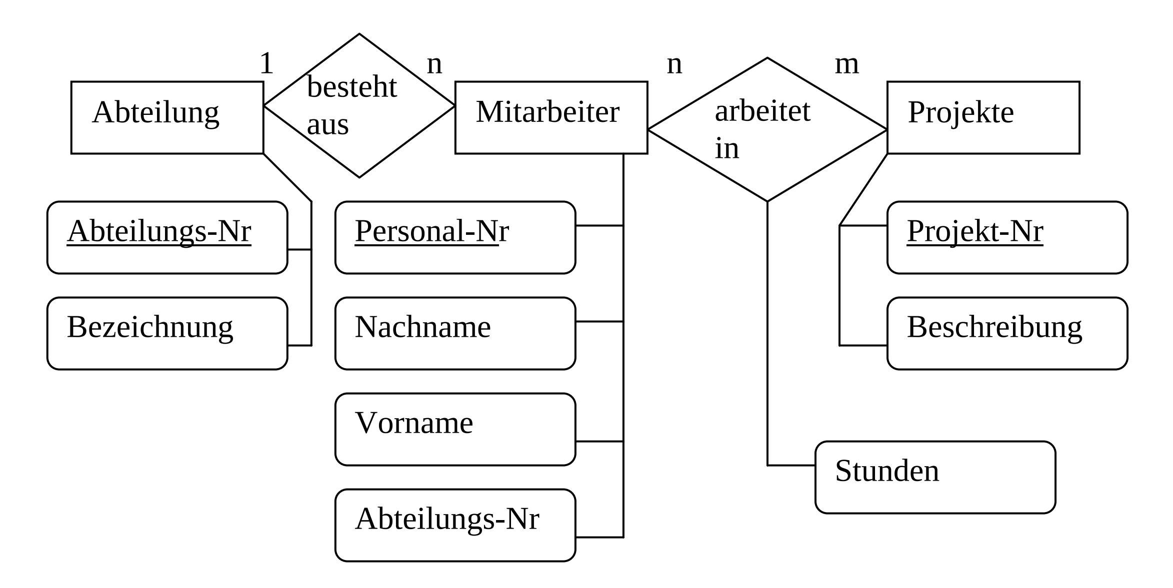
Amtsbez	**Besoldungsgruppe**
StRin	A13
StD	A15
OStR	A14

1	Bavaria	Eusebia	StRin
2	Bachmann	Hanna	StRin
3	Kalkulus	Carl-Johann	StD
4	Spike	Moses	OStR

<u>Nr</u>	**<u>Klasse</u>**
1	11WA
1	13ET
3	11WB
4	12SC
5	12GB

Aufgabe 5

Gegeben sei folgendes ERM.



Geben Sie die sich ergebende relationale Tabellenstruktur in der Kompaktschreibweise "Tabellenname(*PS*, Attribute)" an. Leiten Sie dabei Ihre Lösung ausführlich her und dokumentieren Sie die einzelnen Schritte.

I. Abteilung—besteht aus--Mitarbeiter

Mitarbeiter ist obligatorisch an der Beziehungsrelation beteiligt → mit ihr vereinigen

ergibt

ABTEILUNG(Abteilungs-Nr, Bezeichnung)

MITARBEITER(Personal-Nr, Nachname, Vorname, Abteilungs-Nr) → ist nichts hinzu gekommen, da die Beziehungsrelation keine eigenen Attribute hat. Beziehung zur Abteilung wird über den Fremdschlüssel aufrecht erhalten.

II. Mitarbeiter—arbeitet in--Projekt

n:m → drei Tabellen

MITARBEITER(Personal-Nr, Nachname, Vorname, Abteilungs-Nr)

PROJEKT(Projekt-Nr, Beschreibung)

und die Beziehungsrelation mit den eigenen Attributen und den PS der beteiligten Entitäten als zusammengesetzter PS

MITARBEITERinProjekt(Projekt-Nr, Personal-Nr, Stunden))

Anmerkung:	Relationale Tabellenstrukturen, die aus einem ERM abgeleitet wurden, sind immer mindestens in der 2.Normalform.

Aufgabe 6

Sie importieren Daten aus einer Excel-Tabelle und erhalten die folgende Struktur. Normalisieren Sie sie bis zur 3. Normalform. Dokumentieren Sie dabei ausführlich Ihre Schritte mit der entsprechenden Begründung (Definition der Normalformen und ihre Folgerungen).

PersNr	**Nachname**	**Vorname**	**Abteilung**	**Projekt**	**Stunden**
1	Lorenz	Christian	Einkauf	Verkaufsanalyse	198, 201
2	Baumann	Peter	Verkauf	Optimierung	120, 189, 43
3	Petersen	Anna	Personal	Weiterbildung	120

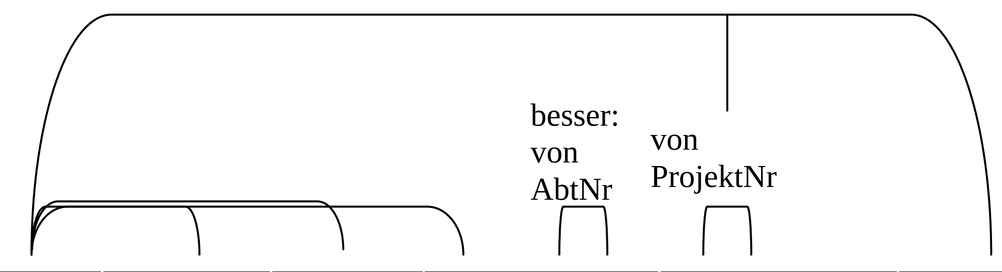
1. Normalform nur atomare Werte erlaubt

PS muss einen Datensatz eindeutig identifizieren → i.d.R. sind dazu zusammengesetzte PS nötig

PersNr	**Nachname**	**Vorname**	**Abteilung**	**Projekt**	**Stunden**
1	Lorenz	Christian	Einkauf	Verkaufsanalyse	198
1	Lorenz	Christian	Einkauf	Verkaufsanalyse	201
2	Baumann	Peter	Verkauf	Optimierung	120
2	Baumann	Peter	Verkauf	Optimierung	189
2	Baumann	Peter	Verkauf	Optimierung	43
3	Petersen	Anna	Personal	Weiterbildung	120

Wir analysieren nun die Tabelle, um aus ihr die 2. und 3. NF ablesen zu können..

von Kombination aus PersNr+ProjektNr



PersNr	**Nachname**	**Vorname**	**Abteilung**	**Projekt**	**Stunden**
1	Lorenz	Christian	Einkauf	Verkaufsanalyse	198
1	Lorenz	Christian	Einkauf	Verkaufsanalyse	201
2	Baumann	Peter	Verkauf	Optimierung	120
2	Baumann	Peter	Verkauf	Optimierung	189
2	Baumann	Peter	Verkauf	Optimierung	43
3	Petersen	Anna	Personal	Weiterbildung	120

2. Normalform Jedes Nicht-Schlüssel-Attribut ist funktional abhängig vom Gesamtschlüssel, nicht dagegen von Schlüsselteilen

Wir haben hier einen zusammengesetzten PS vorliegen, deshalb müssen wir uns überhaupt erst um die 2.NF kümmern, sonst wären die Bedingungen ja schon erfüllt.

Die Stunden hängen nur vom Projekt ab, nicht von der PersNr oder dem Nachnamen. Daher auslagern in eine eigene Tabelle. Gleiches gilt für den Abteilungsnamen.

Demnach

PersNr	Nachname	Vorname	AbteilungsNr	Abteilung
1	Lorenz	Christian	A	Einkauf
2	Baumann	Peter	B	Verkauf
3	Petersen	Anna	B	Personal

PERSONAL

Anmerkung: “AbteilungsNr“ ist streng genommen für die Normalformen hier niicht ötig, jedoch dienen die Normalformen ja der Vermeidung von Inkonsistenzen und Redundanzen. Also führt man sie tunlichst ein und gliedert die beiden Objektbereiche “Personaldaten“ und “Abteilung“ in eigene Tabellen aus.

ProjektNr	Projekt
I	Verkaufsanalyse
II	Optimierung
III	Weiterbildung

PROJEKTE

ProjektNr	PersNr	Stunden
I	1	198
I	1	201
II	2	120
II	2	189
II	2	43
III	3	120

ARBEITENan

schafft Verbindungen zwischen den Tabellen

3. Normalform Es gibt keine funktionalen Abhängigkeiten zwischen Nicht-Schlüssel-Attributen, es gibt also keine transitiven Abhängigkeiten.

Die Abteilung ist nur von der “AbteilungsNr“ abhängig, welche wiederum von der “PersNr“ abhängig ist. Damit ist die “Abteilung“ transitiv Abhängig von der “PersNr“ und wird ausgegliedert.

PERSONAL(PersNr, Nachname, Vorname, AbteilungsNr)

→ AbteilungsNr für Verbindung nicht vergessen!

ABTEILUNG(AbteilungsNr, Abteilung)

PROJEKTE(ProjektNr, Projekt)

ARBEITETan(ProjektNr, PersNr, Stunden)

Aufgabe 7

Reflektieren Sie nochmals über das systematische Vorgehen während des Normalisierungsprozesses. Dazu studieren Sie vorliegenden Text und ergänzen die sich ergebenden Tabellen.

Anmerkung: Es gibt mehrere denkbare Lösungen, also mehrere Tabellenstrukturen. Falls Ihre Idee nicht in die vorgesehenen Tabellen passt ergänzen Sie einfach die Spaltenanzahl, fragen sich aber kritisch, ob Ihre Lösung den gestellten Anforderungen genügt. Das primäre Ziel der Aufgabe ist die nochmalige intensive Reflexion des systematischen Normalisierungsprozesses.

Ein Bücherwurm Karlo möchte Ordnung in sein ganz privates Bücherchaos bringen, da ihm die Unmengen an über die Jahre erworbenen Büchern mittlerweile ein wenig über den Kopf gewachsen sind. Entsprechend entschließt er sich spontan eine Datenbank (DB) dafür zu entwickeln. Seine Freundin studiert gerade fleißig Informatik und kommt am Wochenende vorbei, da möchte unser edler Gesell' natürlich ein wahres Kunstwerk mit allen Schikanen vorweisen können.

Als Regeln für gutes Datenbank-Design findet er folgende Weisheiten

- Tabellen sollten keine redundanten (sich wiederholende) Daten enthalten. Wenn in einer Tabelle immer wieder dasselbe eingegeben werden muss, stimmt etwas nicht.
- Bei der Konzeptentwicklung der DB (konzeptuelles Modell) bietet sich z.B. Excel an, da man damit leicht die Tabellen und ihre Attribute visualisieren kann, ohne auf die spezifischen Restriktionen eines echten DBMS achten zu müssen

Diese Regeln sind formalisiert als Normalisierung bekannt.

Als erster Schritt im DB-Design ist der Weltenausschnitt zu modellieren, den man gerade betrachtet. Bei einer Bücherdatenbank sind entsprechend die Attribute zu ermitteln, die in der DB verwaltet werden sollen. Es wären dies beispielsweise

- Titel (title)
- Verlag (publName)
- Veröffentlichungsjahr (year)
- Autoren (authorX)

Bei der Ermittlung dieser Attribute helfen Realweltdaten wie Karteikarten, Akten usw. Eine handschriftliche Bücherliste von Karlo hat beispielsweise folgende Struktur:

title	publName	year	author1	author2	author3
Linux	Addison-Wesley	2004	Meyer		
Web	Apress	2003	Schulz	Einstein	
Mathe	New Riders	2006	Schmitz		
Physik	Einstein-Verlag	1901	Newton	Schrödinger	Heisenberg
Lyrik	Federweiß	2006	Schulz		

Nun ja, also genau genommen hatte bei der Physik auch noch Max Planck seine Finger im Spiel, aber es war halt nur Platz für drei Autoren… Nicht sehr schön. Die Festlegung auf drei Autoren ist willkürlich, unflexibel und schlichtweg unschönes Design. Auch die Lösung alle Autoren in ein einziges Feld authors zu schreiben

authors
Newton, Schrödinger, Heisenberg, Planck

ist nicht so das wahre, da Karlo gehört hat man solle pro Attribut, also pro Spalte in einer Tabelle, nur *atomare Werte*, also ein Datum statt mehrerer, verwalten.
Nun denn, als Designkriterien für diese erste Hürde gelten folgende Weisheiten , die der *ersten Normalform*:

- Nur atomare Werte pro Spalte.
- Spalten mit gleichartigem Inhalt müssen entfernt werden.
- Für jede Gruppe zusammengehöriger Daten muss eine eigene Tabelle gebildet werden.
- Jeder Datensatz muss durch einen sog. Primärschlüssel eindeutig identifiziert werden.

Die dritte Regel ist schon erfüllt, da momentan in jeder Spalte nur Daten gesammelt werden, die auch zu einem Buch gehören. Regel 1 ist gleichfalls erfüllt, da das Problem der Mehrfachdaten pro Attribut von Karlo durch die drei Autorenspalten gelöst wurde, auch wenn sich dadurch Probleme mit Regel zwei eingefangen hat. Regel 4 ist leicht zu befolgen. Es wird einfach eine fortlaufende Nummer eingeführt (AutoWert), die *titleID*, und als Primärschlüssel verwendet. Diese Nummer identifiziert dann das Buch mit all seinen Attributen eindeutig. Entsprechend ergibt sich die Tabelle nun wie folgt:

titleID	title	publName	year	author
1	Linux	Addison-Wesley	2004	Meyer
2	Web	Apress	2003	Schulz
3	Mathe	New Riders	2006	Schmitz
4	Physik	Einstein-Verlag	1901	Newton
5	Physik	Einstein-Verlag	1901	Schrödinger
6	Physik	Einstein-Verlag	1901	Heisenberg
7	Physik	Einstein-Verlag	1901	Planck
8	Lyrik	Federweiß	2006	Schulz

Prima, alle Kriterien der ersten Normalform sind erfüllt! Selbst das Autorenanzahlproblem wurde gelöst. Allerdings mag es doch verdächtig anmuten alle Daten für n-Autoren n-mal vorrätig zu halten. Das kann nicht der Weisheit letzter Schluss sein.
Eine Tabelle wird in die *zweite Normalform* umgeformt, in dem

- Eine Tabelle in erster Normalform ist.
- Immer dann, wenn sich Inhalte in Spalten wiederholen, müssen die Tabellen in mehrere Teiltabellen zerlegt werden.
- Die Tabellen müssen durch sog. Fremdschlüssel miteinander verbunden werden.

In obiger Tabelle wiederholen sich Daten auf Grund der Autoren. Eine Auslagerung derselbigen ergibt folgende Tabellenstruktur:

titleID	title	publName	year
1	Linux	Addison-Wesley	2004
2	Web	Apress	2003
3	Mathe	New Riders	2006
4	Physik	Einstein-Verlag	1901
5	Lyrik	Federweiß	2006

authID	titleID	author
1	1	Meyer
2	2	Schulz
3	3	Schmitz
4	4	Newton
5	4	Schrödinger
6	4	Heisenberg
7	4	Planck
8	5	Schulz

Schon besser. Allerdings kommt 'Schulz' als Autor immer noch mehrmals vor. Dies tritt immer dann auf, wenn ein Autor mehr als ein Buch geschrieben hat. Als Lösung käme eine weitere Aufteilung der Tabelle in Frage:

titles

titleID	title	publName	year
1	Linux	Addison-Wesley	2004
2	Web	Apress	2003
3	Mathe	New Riders	2006
4	Physik	Einstein-Verlag	1901
5	Lyrik	Federweiß	2006

authors

authID	author
1	Meyer
2	Schulz
3	Schmitz
4	Newton
5	Schrödinger
6	Heisenberg
7	Planck

rel_title_author

titleID	**authID**
1	1
2	2
3	3
4	4
4	5
4	6
4	7
5	2

Letztere Tabelle muss von Rechts nach Links gelesen werden. Sie gibt also an, welche Bücher ein Autor – identifiziert über seine eindeutige authID – geschrieben hat. Vom Buch in einer Anfrage geht es also über die titleID zur authID, welche schließlich zu den Autoren des Buches führt. Die Tabelle rel_title_author hat übrigens einen *zusammengesetzten Primärschlüssel*, der aus den beiden Attributen besteht. Er identifiziert jedes Tupel eindeutig, daher ist ein zusätzlicher Primärschlüssel nicht nötig.
Es gibt noch eine weitere wichtige Normalform, die *dritte Normalform*. Diese löst indirekte (*transitive*) Abhängigkeiten auf. Ist A von B abhängig und B von C, so ist A transitiv von C abhängig, was nicht erlaubt ist, wenn eine Tabelle der dritten Normalform genügen soll. Ein typisches Beispiel ist die Postleitzahl und der Ort. In einer Mitarbeitertabelle werden die Personalstammdaten inklusive Wohnort erfasst. Eine MitarbeiterID (A) identifiziert den Mitarbeiter eindeutig, auch die zugehörige PLZ (B). Der Wohnort (C) ist aber abhängig von der PLZ (B) und damit besteht transitive Abhängigkeit zwischen MitarbeiterID und Wohnort. Entsprechend werden PLZ und Ort aus der Tabelle in eine eigene ausgegliedert.

PersID	Name	…	PLZ	Ort
1			33433	Marl
2			33453	Gelsenkirchen
3			2432	New York

→

PersID	Name	…	PLZ
1			33433
2			33453
3			2432

PLZ	Ort
33433	Marl
33453	Gelsenkirchen
2432	New York

Entsprechend reagiert Karlo und findet heraus, dass die Spalte *publName* in der *title*-Tabelle betroffen ist. Die Liste der Verlage und die Liste der Buchtitel sind voneinander abhängig und sollten daher getrennt behandelt werden. Es ist zwar nötig und sinnvoll zu jedem Buch den Verlag zu vermerken, aber es reicht doch ein kurzer Schlüssel statt immer der vollständigen Daten (hohen Fehleranfälligkeit und große Speicherplatzverschwendung, was zudem den Zugriff auf die Daten verlangsamt). Dies ist der Hintergrund für die dritte Normalform.
Gesagt, getan. Karlo macht folgendes daraus.

titleID	title	publID	year
1	Linux	1	2004
2	Web	2	2003
3	Mathe	3	2006
4	Physik	4	1901
5	Lyrik	5	2006

publID	publName
1	Addison-Wesley
2	Apress
3	New Riders
4	Einstein-Verlag
5	Federweiß

Die Tabellen *rel_title_author* und *authors* änders sich in der dritten Normalform nicht mehr. Die gesamte Bücherdatenbank besteht nun aus drei Tabellen.

authors

authID	author
1	Meyer
2	Schulz
3	Schmitz
4	Newton
5	Schrödinger
6	Heisenberg
7	Planck

rel_title_author

titleID	authID
1	1
2	2
3	3
4	4
4	5
4	6
4	7
5	2

titles

titleID	title	publID	year
1	Linux	1	2004
2	Web	2	2003
3	Mathe	3	2006
4	Physik	4	1901
5	Lyrik	5	2006

publishers

publID	publName
1	Addison-Wesley
2	Apress
3	New Riders
4	Einstein-Verlag
5	Federweiß

Die Normalformen formalisieren demnach zwei Designprinzipien:

- Fasse in einer Tabelle nur das zusammen, was zusammen gehört.
- Vermeide Redundanzen und Abhängigkeiten durch Separierung in eine eigenen Tabelle.

Erweiterungen:

Die Datenbank könnte noch leicht um einige weitere – realistische – Features erweitert werden. So könnte die *titles*-Tabelle noch weitere Felder erhalten, wie etwa einen *Untertitel* oder einen *Kommentar*. Eine weitere languages-Tabelle könnte eine Liste aller Sprachen enthalten, in denen das Buch erschienen ist. Ein Verweis darauf könnte über ein *langID*-Feld in der Tabelle *titles* erfolgen. Eine Tabelle *categories* könnte eine Liste der Buchkategorien enthalten, die hierarchisch gegliedert ist, z.B. 'computer books → databases' und 'computer books → programming', 'computer books → programming → java' usw. Erfasst werden könnte diese Kategorie-Datailtabelle über ein Feld *catID* in der Tabelle *titles*.

Exkurs: Hierarchien

Categories ordnet Bücher in verschiedene Kategorien, die hierarchisch strukturiert sind. Dabei verweist *parentID* auf die übergeordnete Ebene.

All Books
- Children books
- Computer books
  - Databases
    - Object oriented db
    - Relational db
    - Sql
  - Programming
    - Pearl
    - Php

→

catID	catName	parentID
1	computer books	11
2	databases	1
3	programming	1
…		
11	all books	11

Aufgabe 8

Eine Möbelfirma stellte vom Karteikartensystem

Artikelbezeichnung: ***Gabriele***		
Artikelart: ***Schreibtisch***	Artikelnummer: ***1000***	
Bestand:	Mindestbestand: ***5***	Höchstbestand: ***10***
Einkaufspreis:	Verkaufspreis:	
Lieferant: ***Tranel GmbH***	Name2: ***Büromöbel***	
Lieferer_Nr: ***102***	Anschrift: ***Bechemstr. 67, 47058 Duisburg***	

auf Excel um und möchte nun – mit steigendem Datenaufkommen – eine echte Datenbank nutzen. Es gibt sich nach dem Import folgende Tabellenstruktur. Normalisieren Sie sie.

Artikel_Nr	Artikelart	Artikel_Bez	LiefererNr	Name1	Name2	Anschrift
1000	Schreibtisch	Gabriele	102	Tranel GmbH	Büromöbel	Bechemstr., 33242 Duisburg
1001	Schreibtisch	Modern	101	Büromöbel AG	Büroeinrichtungen	Gutachtstr. 342, 12345 Bremen
1002	Schreibtisch	Exklusiv	102	Tranel GmbH	Büromöbel	Vogtweg 34, 32453 Verlin
1003	Büroschrank	Elegant	100	Wagner GmbH	Büromöbel	Becherstrasse 4, 34343 Oberhuisen
1004	Büroschrank	Aktuell	101	Büromöbel AG	Büroeinrichtungen	Gassengosse 2, 36999 Schwerten

0. Normalform

- nichtatomare Feldinhalte
- Lieferanten teils mehrfach erfasst → Mehrarbeit + Dateninkonsistenzgefahr

also

1. Normalform

- **Wiederholungsgruppen eliminieren**
- **Für atomare Werte sorgen**

Attributwert nicht atomar.

Anschrift
Bechemstr., 33242 Duisburg
Gutachtstr. 342, 12345 Bremen
Vogtweg 34, 32453 Verlin
Becherstrasse 4, 34343 Oberhuisen
Gassengosse 2, 36999 Schwerten

Das Datenfeld Anschrift ist daher in drei Tabellenfelder mit eindeutigen Informationen zu teilen.

Straße	PLZ	Ort
Bechemstr.	33242	Duisburg
Gutachtstr.	12345	Bremen
Vogtweg 34	32453	Verlin
Becherstrasse 4,	34343	Oberhuisen
Gassengosse 2	36999	Schwerten

Besser: für die Hausnummer ein eigenes Attribut vorsehen

Artikel_Nr	Artikelart	Artikel_Bez	LiefererNr	Name1	Name2	
1000	Schreibtisch	Gabriele	102	Tranel GmbH	Büromöbel	→
1001	Schreibtisch	Modern	101	Büromöbel AG	Büroeinrichtungen	
1002	Schreibtisch	Exklusiv	102	Tranel GmbH	Büromöbel	→
1003	Büroschrank	Elegant	100	Wagner GmbH	Büromöbel	
1004	Büroschrank	Aktuell	101	Büromöbel AG	Büroeinrichtungen	→

Straße	PLZ	Ort
Bechemstr.	33242	Duisburg
Gutachtstr.	12345	Bremen
Vogtweg 34	32453	Verlin
Becherstrasse 4,	34343	Oberhuisen
Gassengosse 2	36999	Schwerten

Keine Mehrfachattribute/Wiederholungsattribute (z.B. Liefer1,Datum1, Liefer2,Datum2, …) hier ok: ☺

Ein Artikel kann auch von mehreren Lieferanten geliefert werden. Daher ist ein zusammengesetzter PS nötig, um einen Datensatz eindeutig zu identifizieren. Wir wählen Artikel_Nr und LiefererNr.

2. Normalform

- **Attribute müssen vom gesamten PS abhängig sein, nicht schon von Teilend des PS**
- **Besteht der PS aus nur einem Attribut ist die 2 NF automatisch gegeben**

Es ergibt sich in der 2. NF die logische Aufteilung in Domänen. So werden Lieferantendaten in der Tabelle LIEFERANTEN gespeichert und Artikeldaten in der Tabelle LAGER.

In der Tabelle Lieferanten werden alle Lieferantendaten gespeichert und über den PS LiefererNr eindeutig identifiert

<u>LiefererNr</u>	Name1	Name2	PLZ	Ort
102	Tranel GmbH	Büromöbel	33242	Duisburg
101	Büromöbel AG	Büroeinrichtungen	12345	Bremen
102	Tranel GmbH	Büromöbel	32453	Verlin
100	Wagner GmbH	Büromöbel	34343	Oberhuisen
101	Büromöbel AG	Büroeinrichtungen	36999	Schwerten

In der Tabelle Lager werden alle Artikeldaten erfasst. Als Primärschlüssel wird die Spalte Artikel_Nr bestimmt. Alle anderen Spalten sind abhängig von dieser Spalte.

<u>Artikel_Nr</u>	Artikelart	Artikel_Bez	Bestand	Einkaufspreis
1000	Schreibtisch	Gabriele	45	60
1001	Schreibtisch	Modern	2	65
1002	Schreibtisch	Exklusiv	0	120
1003	Büroschrank	Elegant	19	100
1004	Büroschrank	Aktuell	24	250

(Bestand, Einkaufspreis nur zur Verdeutlichung weiterer sinnvoller Attribute mit aufgeführt. Kommen ja in der Ausgangstabelle gar nicht vor)

3 Normalform

- **tansitive (indirekte) Abhängigkeiten auflösen**

Die Tabelle ARTIKEL erfüllt das Kriterium schon. Die LIEFERANTEN Tabelle weist allerdings den Klassiker transitiver Abhängigkeit auf, die Abhängigkeit des *Ortes* von der *PLZ* und nicht vom PS L*ief*ererNr. Entsprechend entstehen zwei Tabellen.

<u>LiefererNr</u>	Name1	Name2	PLZ
102	Tranel GmbH	Büromöbel	33242
101	Büromöbel AG	Büroeinrichtungen	12345
102	Tranel GmbH	Büromöbel	32453
100	Wagner GmbH	Büromöbel	34343
101	Büromöbel AG	Büroeinrichtungen	36999

<u>PLZ</u>	Ort
33242	Duisburg
12345	Bremen
32453	Verlin
34343	Oberhuisen
36999	Schwerten

Insgesamt erhalten wir also folgendes Ergebnis:

<u>LiefererNr</u>	Name1	Name2	PLZ
102	Tranel GmbH	Büromöbel	33242
101	Büromöbel AG	Büroeinrichtungen	12345
102	Tranel GmbH	Büromöbel	32453
100	Wagner GmbH	Büromöbel	34343
101	Büromöbel AG	Büroeinrichtungen	36999

<u>PLZ</u>	Ort
33242	Duisburg
12345	Bremen
32453	Verlin
34343	Oberhuisen
36999	Schwerten

Artikel_Nr	Artikelart	Artikel_Bez	Bestand	Einkaufspreis
1000	Schreibtisch	Gabriele	45	60
1001	Schreibtisch	Modern	2	65
1002	Schreibtisch	Exklusiv	0	120
1003	Büroschrank	Elegant	19	100
1004	Büroschrank	Aktuell	24	250

Aufgabe 9

Ein Betrieb mit Außendienstmitarbeitern erfasst ihre Reisekosten in einer Tabelle, die wie folgt aufgebaut ist.

Reisekosten									
Rechnungs-nummer	Datum	Name	Vorname	Straße	PLZ	Ort	Kostenart	Anzahl	Einzelvergütung

Normalisieren Sie sie und dokumentieren Sie ausführlich Ihre jeweiligen Schritte mit Verweis auf die Fachbegriffe.

Überlegen Sie sich am Ende ferner, welche Verbindungen zwischen welchen Tabellen mit entsprechender Kardinalität auftreten könnten.

1. Normalform

Ok, alle Attribute sind schon atomar und Mehrfachattribute/Wiederholungsattribute gibt es nicht. Also keine Änderungen nötig.

2. Normalform

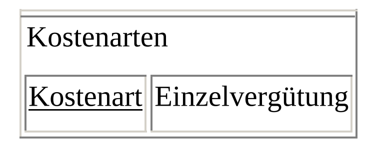
Es gibt hier einen zusammengesetzten PS, also ist die 2.NF überhaupt ein Thema. Auch hier liefert das formale Vorgehen wieder das logische Ergebnis, welches man auch erhalten hätte, wenn man „im Kopf" die Attribute nach Zusammengehörigkeit geordnet hätte. Es ist volle funktionale Abhängigkeit vom GESAMTprimärschlüssel gefordert, also müssen wir die Tabellen aufteilen.

Wir stellen fest, dass die *Rechnungsnummer* viele Attribute eindeutig bezeichnet, nämlich *Datum*, *Name*, *Vorname*, *Straße*, *PLZ* und *Ort*.

Die *Einzelvergütung* wird bestimmt von der *Kostenart*, die *Anzahl* vom gesamten PS.

Also ergibt sich folgendes:

Reise						
Rechnungsnummer	Datum	Name	Vorname	Straße	PLZ	Ort



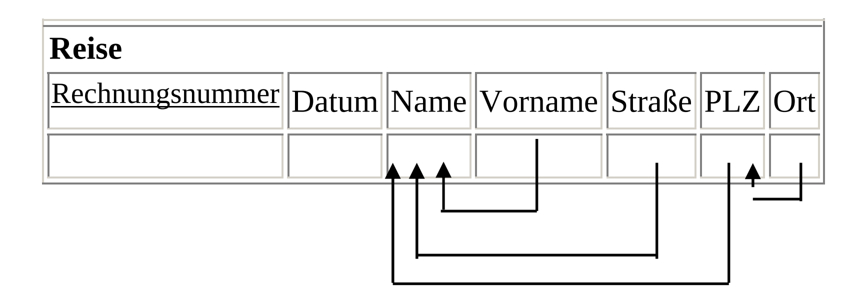
Kostenarten	
Kostenart	Einzelvergütung

Positionen		
Rechnungsnummer	Kostenart	Anzahl

3. Normalform

In der Tabelle "Reise" sind die Attribute *Vorname*, *Straße* und PLZ abhängig vom Attribut *Name*, nicht vom Primärschlüssel. Außerdem ist *Ort* abhängig von *PLZ*. Zu jeder *Rechnungsnummer* gehört eine *PLZ* und zu jeder *PLZ* ein *Ort*, also zu jeder *Rechnungsnummer* ein *Ort*.

Abhilfe:
Die transitiv abhängigen Datenfelder werden in weitere Tabellen ausgelagert, da sie nicht direkt vom Schlüsselkandidaten abhängen, sondern nur indirekt.
Da ein Name die Adresse bestimmt und nicht der momentane PS Rechnungsnummer, wird jedem Angestellten eine Personalnummer zugeordnet und diese Personaldaten auch in einer eigenen Tabelle erfasst. Diese ist Primärschlüssel der neuen Tabelle "Personal". Alternativ könnte ein zusammengesetzter Primärschlüssel aus Name, Vorname und Geburtsdatum benutzt werden (dieser sollte hinreichend eindeutig sein). Dieses Ergebnis erhielte man auch nach der ‚natürlichen“ Aufteilung der Daten in Tabellen, denn die Rechnung hat ja nicht direkt etwas mit der Adresse zu tun, nur indirekt. Es handelt sich logisch betrachtet um unterschiedliche Domänen.



Reise						
Rechnungsnummer	Datum	Name	Vorname	Straße	PLZ	Ort

Personal				
Personalnummer	Name	Vorname	Straße	PLZ

Reise		
Rechnungsnummer	Datum	Personalnummer

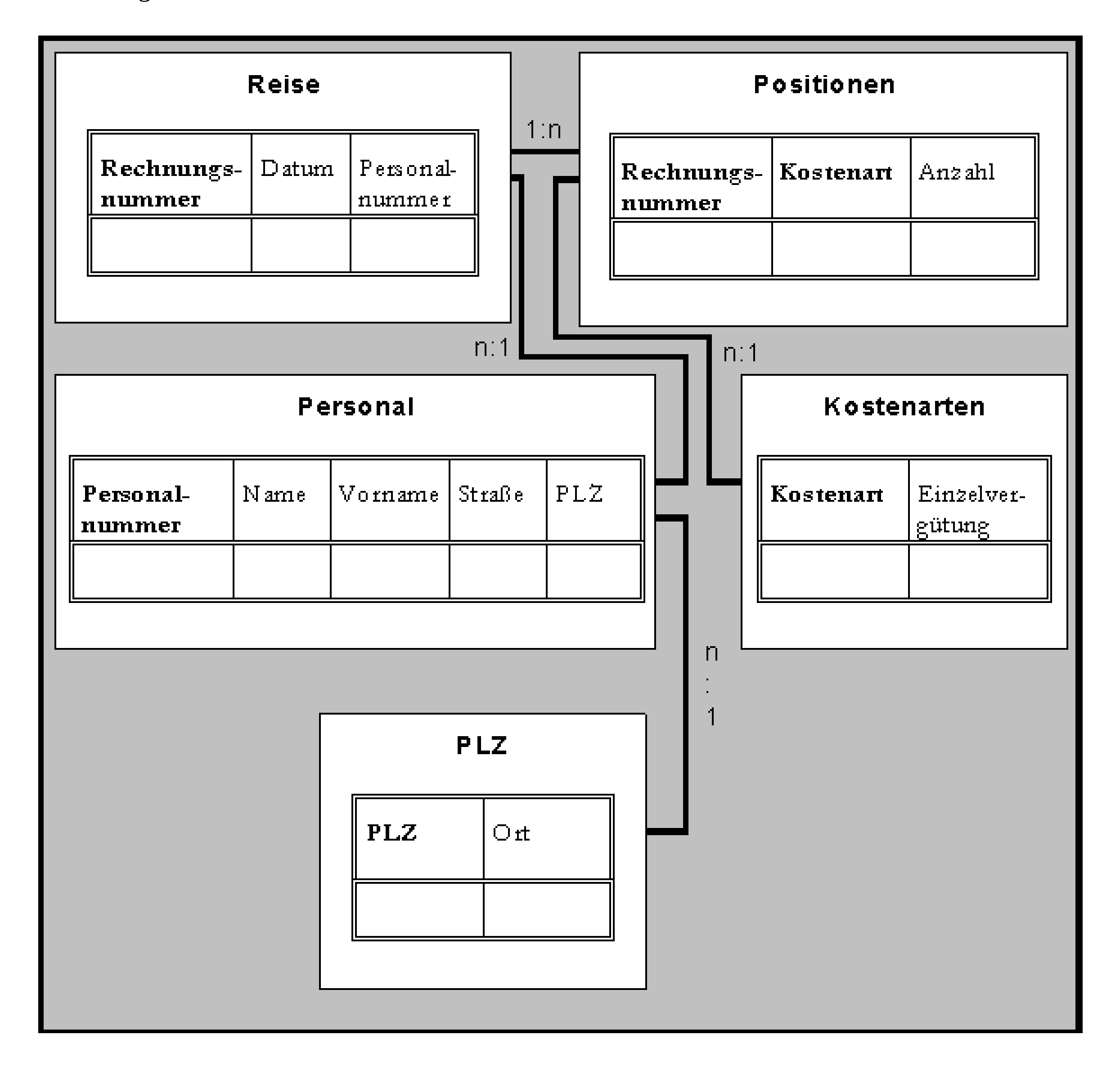
Kostenarten	
Kostenart	Einzelvergütung

Positionen		
Rechnungsnummer	Kostenart	Anzahl

PLZ	
PLZ	Ort

Wiederum besteht zwischen den Tabellen "Personal" und "Reise" sowie zwischen "PLZ" und "Personal" eine 1:n Beziehung, da ein Mitarbeiter mit einer bestimmten Personalnummer mehrere Dienstreisen machen kann und mehrere Mitarbeiter dieselbe Postleitzahl haben können.

Nach der dritten Normalisierung ergeben sich also folgende Tabellen, Datenfelder und Beziehungen:



Beachten Sie:

- Bei der Festlegung der Beziehungen kommt es auf die Reihenfolge der Tabellen an:
  Die Tabelle "PLZ" steht in einer 1:n Beziehung zur Tabelle "Personal", die Tabelle "Personal" dagegen in einer n:1 Beziehung (und nicht 1:n) zur Tabelle "PLZ"

- Eine Relationale Datenbank speichert nicht nur die Daten der Tabellen sondern auch die Beziehungen zwischen den Tabellen

Aufgabe 10

Eine Tabelle sei in Kurzschreibweise wie folgt gegeben:

RECHNUNG(<u>RechNr</u>, KundenNr, Datum, Artikel)

Denken Sie sich exemplarische Datenbestände aus, die der 1. Normalform <u>nicht</u> genügen. Lösen Sie diese dann auf, bilden Sie also die 1. NF der Tabelle.

<u>RechNr</u>	KundenNr	Datum	Artikel
1	2	03.03.2020	2 Apfelbäume
2	5	04.01.2008	2 Birnbäume, 3 Apfelbäume
3	12	01.01.1999	2 Pflaumenbäume, 3 Kirschbäume

Es sind nicht-atomare Attributwerte enthalten, also auflösen:

1. Normalform

<u>RechNr</u>	KundenNr	<u>Position</u>	Datum	Anzahl	Artikel
1	2	1	03.03.2020	2	Apfelbaum
2	5	1	04.01.2008	2	Birnbaum
2	5	2	04.01.2008	3	Apfelbaum
3	12	1	01.01.1999	2	Pflaumenbaum
3	12	2	01.01.1999	3	Kirschbaum

Nun ist ein zusammengesetzter PS nötig, um einen Datensatz eindeutig zu identifizieren. Dazu wurde das Feld Positon hinzugefügt, da ansonsten nur alle 4 Attribute zusammen einen eindeuten PS ergeben hätten.

Es gibt nun einen zusammengesetzten PS, daher ist die 2. NF überhaupt noch zu diskutieren.

Datum und *KundenNr* sind abhängig von der *Rechnungsnummer*, entsprechend muss ausgelagert werden.

2. Normalform

RECHNUNG

<u>RechNr</u>	KundenNr	Datum
1	2	03.03.2020
2	5	04.01.2008
2	5	04.01.2008
3	12	01.01.1999
3	12	01.01.1999

RECHNUNGDETAILS

RechNr	Position	Anzahl	Artikel
1	1	2	2 Apfelbäume
2	1	2	2 Birnbäume
2	2	3	3 Apfelbäume
3	1	2	2 Pflaumenbäume
3	2	3	3 Kirschbäume

Lösungsheft

Kapitel 3
–
Das relationale Modell

Aufgaben

Aufgaben zu Operatoren

Aufgabe 1

Gegeben seien drei Relationen mit den folgenden Tupeln:

Besucht

Gast	Bistro
Hans	Uno
Ede	Uno
Ede	Dos
Ede	Chico
Karl	Dos
Karl	Chico
Heini	Uno

Serviert

Bistro	Getränk
Uno	Wasser
Uno	Kaffee
Dos	Kaffee

Mag

Gast	Getränk
Hans	Wasser
Ede	Wasser
Ede	Kaffee
Karl	Kaffee

a) Bilden Sie Serviert × Mag.

b) Bilden Sie *Serviert* $\bowtie$ *Mag.* Welche Informationen beinhaltet diese Relation?

c) Geben Sie alle Bistros aus, die Getränk servieren, die Karl mag. Überprüfen Sie Ihre Operation in der Relationenalgebra anhand des Beispiels.

d) Geben Sie alle Gäste aus, die mindestens ein Bistro besuchen, die auch das Getränk serviert, das sie mögen. Formulieren Sie die Anfrage mit Operationen der Relationenalgebra.

Aufgabe 2

Gegeben seine folgende Relationen (# ist das Zeichen für *Nummer*):

Lieferanten (L#, LName, Status, Stadt)

Teile (T#, TName, Farbe, Gewicht, Stadt)

Projekte (P#, PName, Stadt)

Lieferungen (L#, T#, P#, Anzahl)

Hierbei bedeutet Stadt einmal die Stadt, in der ein Lieferant sitzt, die Stadt, in der das entsprechende Teil hergestellt wird, bzw. die Stadt, in der ein Projekt stattfindet. Lösen Sie die folgenden Aufgaben durch Operationen aus der Relationenalgebra:

a) Finde Sie alle Lieferungen mit Anzahlen zwischen 300 und 750 und geben Sie alle dazu in der Relation Lieferungen verzeichneten Informationen aus.

b) Geben Sie alle Städte aus, in denen Lieferanten sitzen.

c) Geben Sie alle vorkommenden Paarungen TName, Stadt aus.

d) Finden Sie alle schwarzen Teile. Geben Sie ihre Nummer und ihren Namen aus.

e) Finden Sie alle Lieferanten, die in einer Einzellieferung mehr als 150 Teile geliefert haben. Geben Sie ihren Namen aus.

f) Finden Sie alle Teile, die von Lieferanten in London geliefert wurden. Geben Sie davon die Teilenummer (Teilenamen)aus.

g) Finden Sie alle Orte, in denen sowohl Projekte als auch Lieferanten beheimatet sind.

h) Finden Sie alle Projekte, die mindestens einen Lieferanten für das Projekt im gleichen Ort haben. Geben Sie die Projektnummer aus.

i) Finden Sie alle Teile, die der Lieferant Lux geliefert hat. Geben Sie alle Teilinformationen von diesen Teilen aus.

Aufgabe 3

Gegeben sind die folgenden Tabellen:

Tabelle1

A	B	C
4	2	8
2	2	1
6	7	3

Tabelle2

C	D
8	2
3	6

Tabelle3

B	E
5	3
4	4
5	4

Tabelle4

B	C	D
1	2	1
4	2	1
1	2	9

Führen Sie folgende relationalen Operationen durch und stellen Sie die Ergebnistabelle auf! Beschreiben Sie die Aufgabenstellung mittels der behandelten Symbolik!

a) Selektion von Tabelle1 mit der Bedingung B=2
b) Projektion von Tabelle3 auf E
c) Join Tabelle1 und Tabelle2 nach dem gemeinsamen Attribut C
d) (Selektion von Tabelle 1 mit B>C) vereinigt mit (Selektion von Tabelle1 mit A<5)

Aufgabe 4

Gegeben seien folgende Tabellen

GK-Fach 1

Raum	Fach	Lehrer
137	Mathematik	Müller
221	Deutsch	Schmidt
104	Englisch	Lehmann

GK-Fach 2

Raum	Fach	Lehrer
127	Informatik	Müller
104	Englisch	Lehmann
123	Physik	Paulsen
018	Musik	Schmidt

Themen

Nr.	Thema	Klasse
001	Analysis	12/I
002	Klassik	13/1

Bilden Sie

a) GK-Fach 1 $\cap$ GK-Fach 2

b) GK-Fach 1 $\cup$ GK-Fach 2

c) GK-Fach 1 \ GK-Fach 2

d) Themen x GK-Fach 2

e) σ Name = 'Müller'(Schüler)

f) π Name(Schüler)

g) π Vorname(σ Name = 'Müller'(Schüler))

h) Join(Schüler, Kurs)

Lösungen

Aufgabe 1

a) Bilden Sie Serviert × Mag.

Bistro	Getränk	Gast	Getränk
Uno	Wasser	Hans	Wasser
Uno	Wasser	Ede	Wasser
Uno	Wasser	Ede	Kaffee
Uno	Wasser	Karl	Kaffee
Uno	Kaffee	Hans	Wasser
Uno	Kaffee	Ede	Wasser
Uno	Kaffee	Ede	Kaffee
Uno	Kaffee	Karl	Kaffee
Dos	Kaffee	Hans	Wasser
Dos	Kaffee	Ede	Wasser
Dos	Kaffee	Ede	Kaffee
Dos	Kaffee	Karl	Kaffee

b) weiß unterlegte Tupel

c) $\pi_{\text{Bistro}}\,(\sigma_{\text{Gast = Karl}}\,(\text{Serviert} \bowtie \text{Mag})$ oder effizienter

$\pi_{\text{Bistro}}\,(\text{Serviert} \bowtie \pi_{\text{Getränk}}\,(\sigma_{\text{Gast = Karl}}\,(\text{Mag})))$

1. Tabellen verbinden $\bowtie$
2. auf Karl reduzieren σ
3. Frage beantworten π

d) $\pi_{\text{Gast}}\,(\text{Besucht} \bowtie (\pi_{\text{Bistro, Gast}}\,(\text{Serviert} \bowtie \text{Mag})$

Aufgabe 2

In vielen Fällen wurde versucht, die Größe der Zwischentabellen durch frühzeitige Projektion oder Selektion zu minimieren.

a) $\sigma_{\text{Anzahl} \geq 300 \text{ und Anzahl} \leq 750}\,(\text{Lieferungen})$

b) $\pi_{\text{Stadt}}\,(\text{Lieferanten})$

c) $\pi_{\text{TName, Stadt}}\,(\text{Teile})$

d) $\pi_{\text{T\#, TName}}\,(\sigma_{\text{Farbe = schwarz}}\,(\text{Teile}))$

e) $\pi_{\text{Lname}}\,(\pi_{\text{L\#}}\,(\sigma_{\text{Anzahl} > 150}\,(\text{Lieferungen})) \bowtie \pi_{\text{L\#, Lname}}\,(\text{Lieferanten}))$

f) $\pi_{\text{T\#}}\,(\pi_{\text{L\#}}\,(\sigma_{\text{Stadt = London}}\,(\text{Lieferanten})) \bowtie \pi_{\text{L\#, T\#}}\,(\text{Lieferungen}))$

$\bowtie$ $\bowtie$

π_{TName} ($\pi_{L\#}$ ($\sigma_{Stadt\,=\,London}$ (Lieferanten)) $\pi_{L\#,\,T\#}$ (Lieferungen) Teile)

g) π_{Stadt} (Lieferanten ⋈ Projekte)

h) $\pi_{P\#}$ ($\pi_{P\#,\,L\#}$ (Lieferungen) ⋈ $\pi_{P\#,\,Stadt}$ (Projekte) ⋈ $\pi_{L\#,\,Stadt}$ (Lieferanten))

i) $\pi_{T\#}$ ($\pi_{L\#}$ ($\sigma_{Lname\,=\,Lux}$ (Lieferanten) ⋈ $\pi_{L\#,\,P\#}$ (Lieferungen)) ⋈ Teile

Aufgabe 3

a)

A	B	C
4	2	8
2	2	1

b)

E
3
4

c)

A	B	C	D
4	2	8	2
6	7	3	6

d)

A	B	C
4	2	8
2	2	1
6	7	3

Aufgabe 4

Gegeben seien folgende Tabellen

GK-Fach 1

Raum	Fach	Lehrer
137	Mathematik	Müller
221	Deutsch	Schmidt
104	Englisch	Lehmann

GK-Fach 2

Raum	Fach	Lehrer
127	Informatik	Müller
104	Englisch	Lehmann
123	Physik	Paulsen
018	Musik	Schmidt

Bilden Sie

a) GK-Fach 1 ∩ GK-Fach 2

Der Durchschnitt A ∩ B zweier Tabellen A und B ist die Menge aller Tupel, die sowohl in A als auch in B enthalten sind.

Raum	Fach	Lehrer
104	Englisch	Lehmann

b) GK-Fach 1 ∪ GK-Fach 2

Die Vereinigung A ∪ B zweier Tabellen A und B ist die Menge aller Tupel, die in A oder in B oder in beiden Relationen enthalten sind.

GK-Fach 1

Raum	Fach	Lehrer
137	Mathematik	Müller
221	Deutsch	Schmidt
104	Englisch	Lehmann

GK-Fach 2

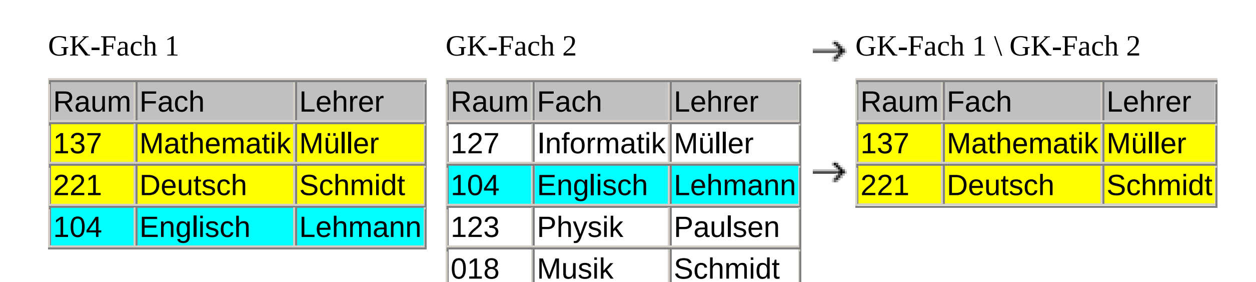
Raum	Fach	Lehrer
127	Informatik	Müller
104	Englisch	Lehmann
123	Physik	Paulsen
018	Musik	Schmidt

→ GK-Fach 1 ∪ GK-Fach 2

Raum	Fach	Lehrer
137	Mathematik	Müller
221	Deutsch	Schmidt
104	Englisch	Lehmann
127	Informatik	Müller
123	Physik	Paulsen
018	Musik	Schmidt

c) GK-Fach 1 \ GK-Fach 2

Die Differenz A \ B zweier Tabellen A und B ist die Menge aller Tupel, die in A aber nicht gleichzeitig in B enthalten sind.



GK-Fach 1

Raum	Fach	Lehrer
137	Mathematik	Müller
221	Deutsch	Schmidt
104	Englisch	Lehmann

GK-Fach 2

Raum	Fach	Lehrer
127	Informatik	Müller
104	Englisch	Lehmann
123	Physik	Paulsen
018	Musik	Schmidt

→ GK-Fach 1 \ GK-Fach 2

Raum	Fach	Lehrer
137	Mathematik	Müller
221	Deutsch	Schmidt

d) Themen x GK-Fach 2

Das Produkt A x B zweier Tabellen A vom Grad n und B vom Grad m ist die Menge aller Kombinationstupel (n x m-Tupel).

Themen

Nr.	Thema	Klasse
001	Analysis	12/I
002	Klassik	13/1

GK-Fach 2

Raum	Fach	Lehrer
127	Informatik	Müller
104	Englisch	Lehmann
123	Physik	Paulsen
018	Musik	Schmidt

→

Nr.	Thema	Klasse	Raum	Fach	Lehrer
001	Analysis	12/I	127	Informatik	Müller
001	Analysis	12/I	104	Englisch	Lehmann
001	Analysis	12/I	123	Physik	Paulsen
001	Analysis	12/I	018	Musik	Schmidt
002	Klassik	13/1	127	Informatik	Müller
002	Klassik	13/1	104	Englisch	Lehmann
002	Klassik	13/1	123	Physik	Paulsen
002	Klassik	13/1	018	Musik	Schmidt

e) σ Name = 'Müller'(Schüler)

Bei der Selektion werden Zeilen aus einer Tabelle ausgewählt, die bestimmten Eigenschaften genügen.

Schüler

SNr	Vorname	Name
4711	Paul	Müller
0815	Erich	Schmidt
7472	Sven	Lehmann
1234	Olaf	Müller
2313	Jürgen	Paulsen

→

σ Name = 'Müller'(Schüler)

SNr	Vorname	Name
4711	Paul	Müller
1234	Olaf	Müller

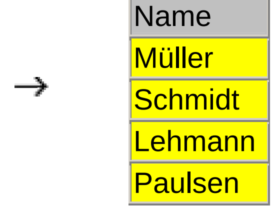
f) π Name(Schüler)

Bei der Projektion werden Spalten aus einer Tabelle ausgewählt, die bestimmten Eigenschaften genügen. Ist ein Eintrag mehrfach vorhanden, so wird er nur einmal angezeigt.

Schüler

SNr	Vorname	Name
4711	Paul	Müller
0815	Erich	Schmidt
7472	Sven	Lehmann
1234	Olaf	Müller
2313	Jürgen	Paulsen

π Name(Schüler)



Name
Müller
Schmidt
Lehmann
Paulsen

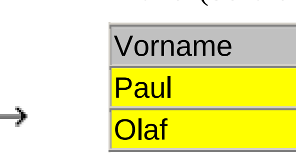
g) π Vorname(σ Name = 'Müller'(Schüler))

Aus der Tabelle Schüler sollen die Vornamen aller Schüler angezeigt werden, deren Nachname Müller ist. Die Abfrage hat also die Form: π Vorname(σ Name = 'Müller'(Schüler))

Schüler

SNr	Vorname	Name
4711	Paul	Müller
0815	Erich	Schmidt
7472	Sven	Lehmann
1234	Olaf	Müller
2313	Jürgen	Paulsen

π Vorname(σ Name = 'Müller'(Schüler))



Vorname
Paul
Olaf

h) Join(Schüler, Kurs

Ein Join ist das Verbinden von zwei Relationen zu einer neuen Tabelle.

Die Literatur unterscheidet eine größere Anzahl Joins, für die Schule sind nur die mit

Schüler

SNr	Vorname	Name
4711	Paul	Müller
0815	Erich	Schmidt
7472	Sven	Lehmann
1234	Olaf	Müller
2313	Jürgen	Paulsen

Kurse

SNr	KNr	Fehlstunden	Punkte
0815	03	0	12
4711	03	12	03
1234	23	3	14
0987	09	9	09

→

Join_{SNr}(Schüler, Kurs)

SNr	Vorname	Name	KNr	Fehlstunden	Punkte
0815	Erich	Schmidt	03	0	12
4711	Paul	Müller	03	12	03
1234	Olaf	Müller	23	3	14

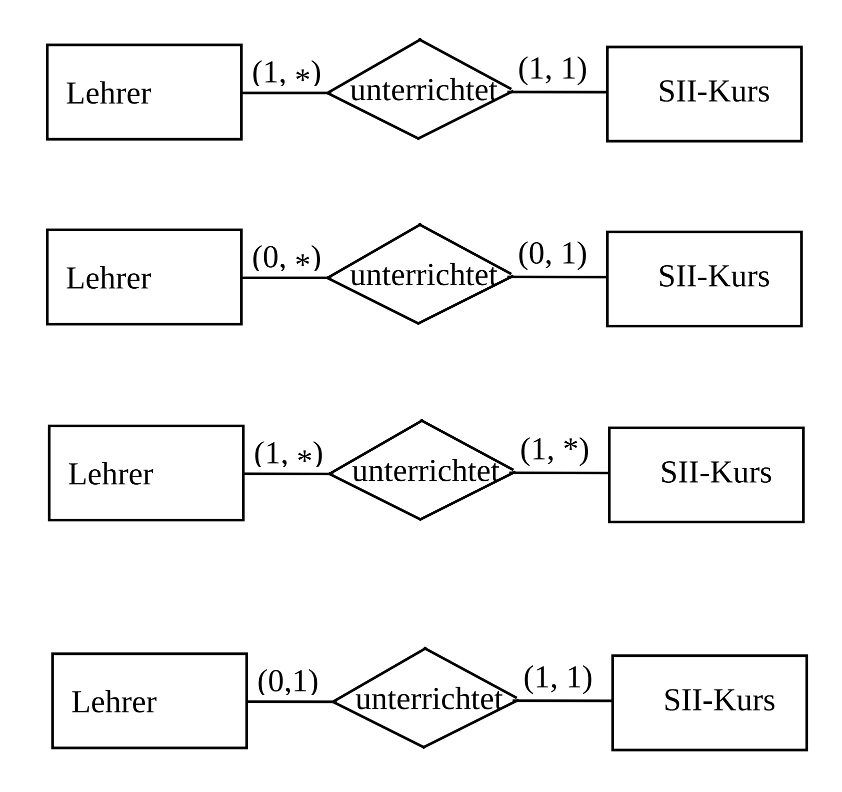
Da zu den SNr.7472, 2314 und 0987 der Tabelle Schüler bzw. Kurse keine "Gegenstücke" in der anderen Tabelle existieren, werden diese in der neuen Relation nicht angezeigt.

Aufgaben

Aufgaben zur Umsetzung in das relationale Modell

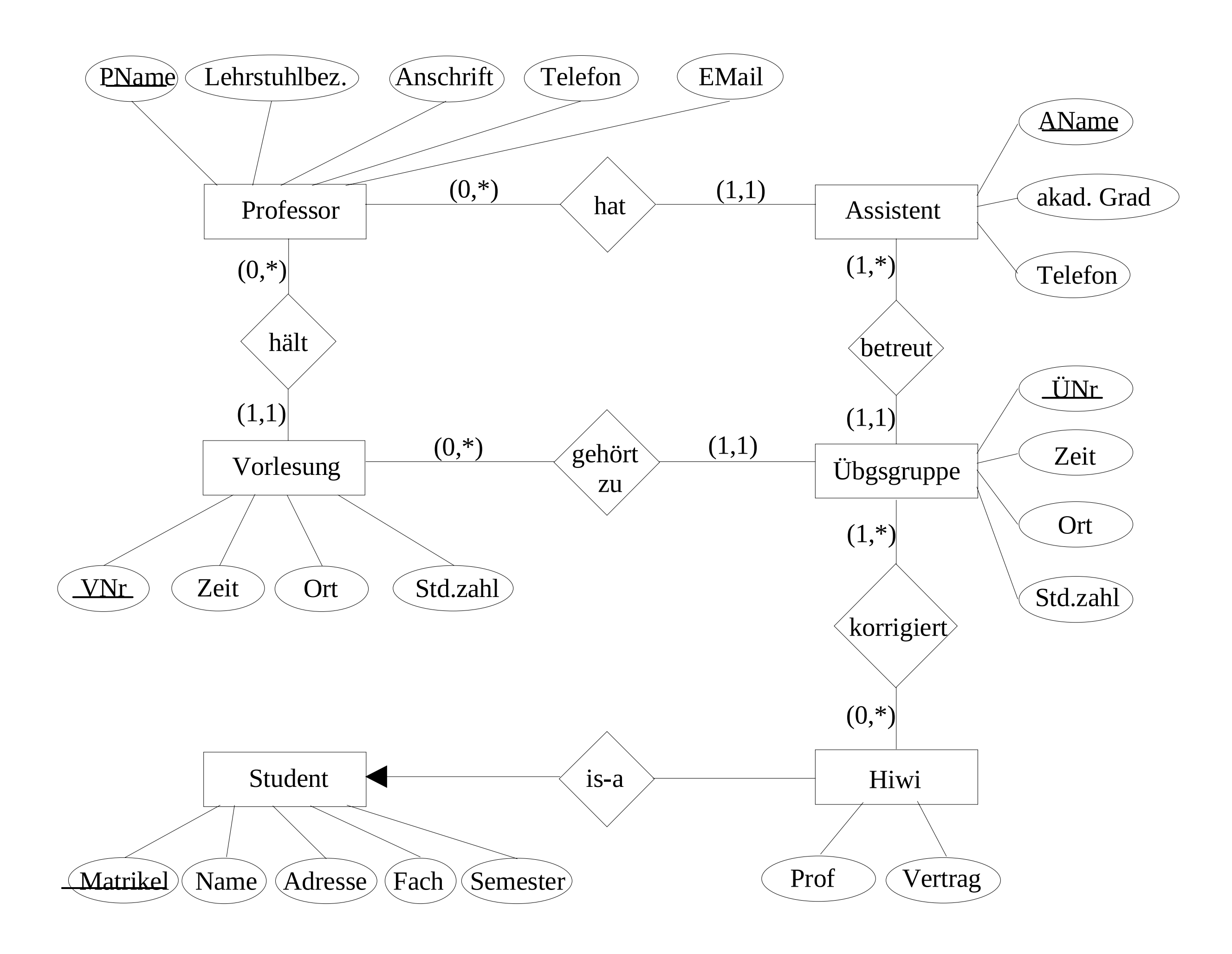
Aufgabe 1

Was bedeuten die folgenden vier Diagramme (Prosa-Erklärung)? Die Attribute wurden zur Vereinfachung weggelassen. Geben Sie jeweils die Komplexität in der Form 1:1, 1:n bzw. n:m an.



Aufgabe 2

Bilden Sie dieses ER-Diagramm in das relationale Modell ab.



Aufgabe 3

Setzen Sie das ER-Diagramm Schule vollständig in ein relationales Modell um, indem Sie alle benötigten Tabellen mit ihren Köpfen angeben, die Primärschlüssel kennzeichnen und durch Pfeile zwischen den Attributen verschiedener Tabellen die Beziehungen verdeutlichen. Im Text wurden schon einige Tabellen eingeführt.

Aufgabe 4

Setzen Sie die unter “Modellierung von Ausschnitten der Wirklichkeit“ dargestellte Aufgabe der Firma Microtec in das relationale Modell um, indem Sie das ER-Modell nach den beschriebenen Verfahren in Tabellen umsetzen!

Aufgabe 5

Setzen Sie das ER-Modell der Spedition Hugo Unbedarft aus „Aufgaben zum ERM“ in ein relationales Modell um. Überlegen Sie dabei, welche Tabellen entstehen müssen und welche Beziehungen sich integrieren lassen.

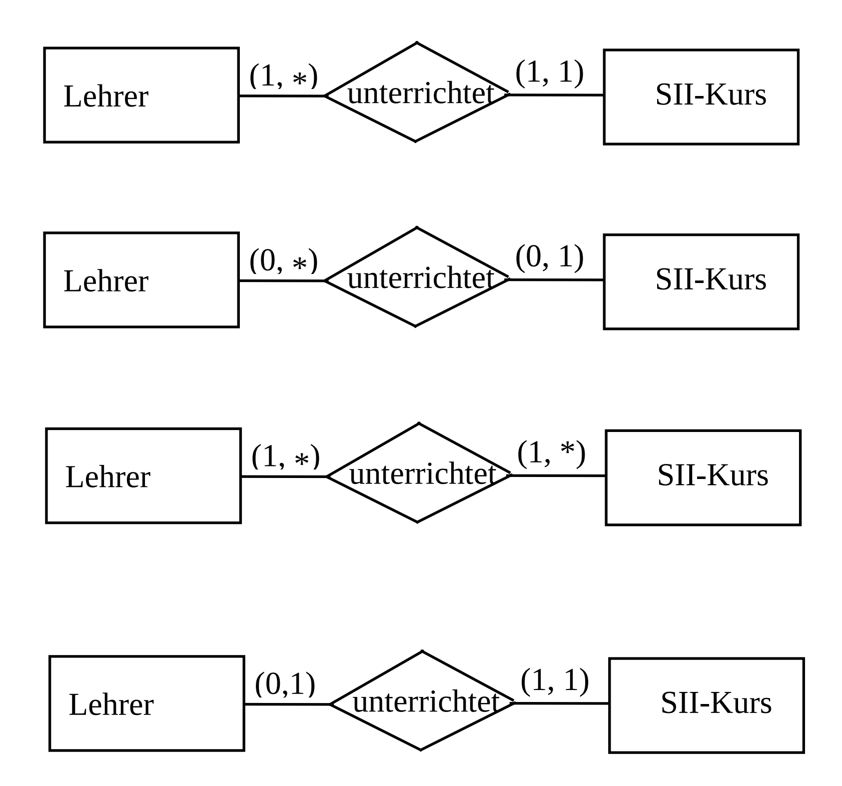
Anmerkung:
Wenn Sie die Normalisierung erlernt haben kommen sie auf diese Aufgabe zurück und Überprüfen anschließend, inwieweit sich die entstandenen Tabellen in der 3. Normalform befinden! Treten Abhängigkeiten zwischen den Schlüsseln auf?

Lösungen

Aufgaben zur Umsetzung in das relationale Modell

Aufgabe 1

Was bedeuten die folgenden vier Diagramme (Prosa-Erklärung)? Die Attribute wurden zur Vereinfachung weggelassen. Geben Sie jeweils die Komplexität in der Form 1:1, 1:n bzw. n:m an.



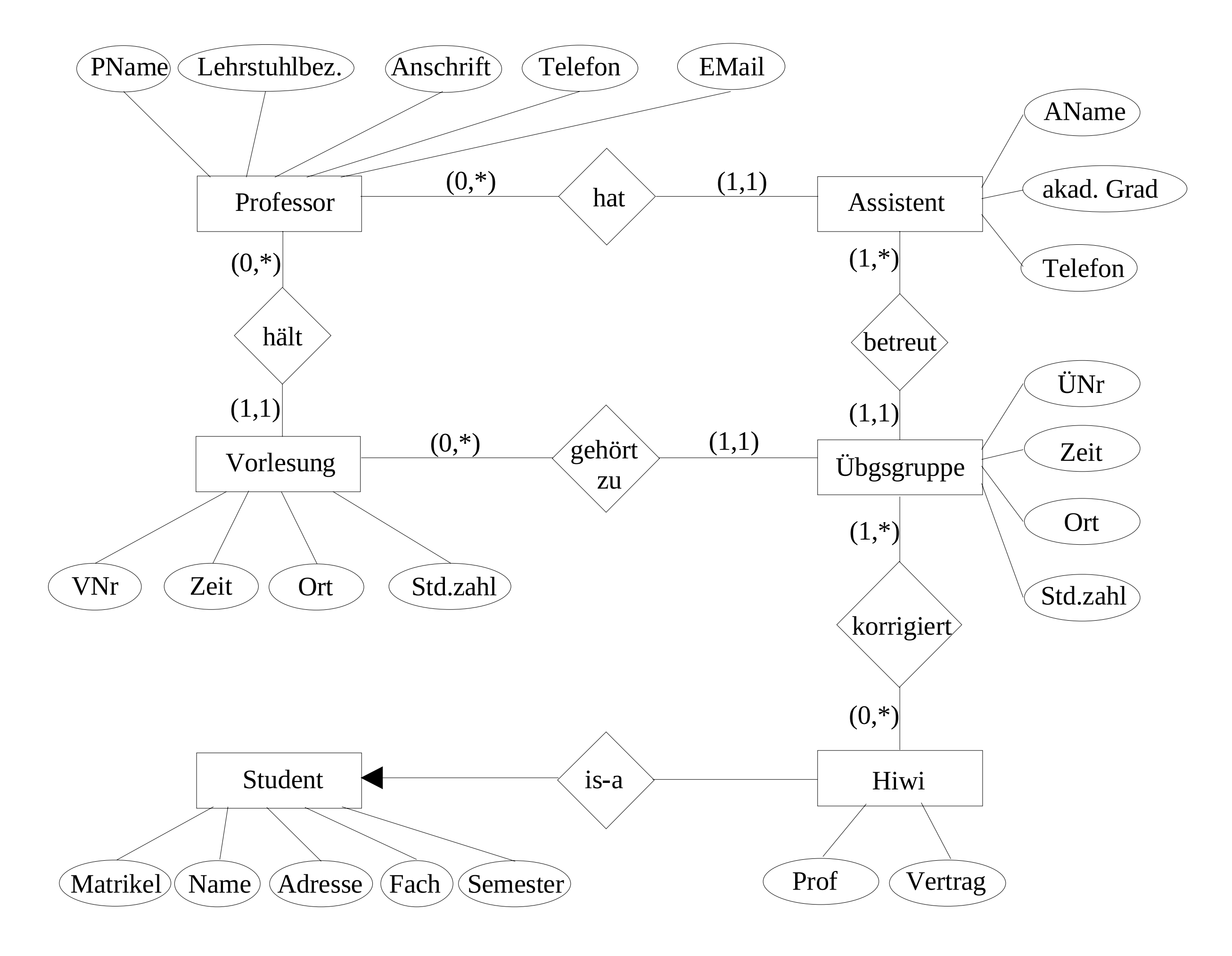
a) Jeder Lehrer hat mindestens einen Kurs, und jeder Kurs hat genau einen Lehrer. 1:n
b) Es gibt auch Kurse ohne Lehrer und Lehrer ohne Kurs. Ein Kurs hat höchstens einen Lehrer. 1:n
c) Jeder Lehrer hat mindestens einen Kurs, und ein Kurs kann mehrere Lehrer haben. n:m
d) Jeder Lehrer hat höchstens einen Kurs, und ein Kurs hat genau einen Lehrer. 1:1

Aufgabe 2

Bilden Sie dieses ER-Diagramm in das relationale Modell ab.

Professor(<u>PName</u>, Lehrstuhlbez., Anschrift, Telefon, EMail)
Assistent(<u>AName</u>, akad. Grad, Telefon, *PName*)
Vorlesung(<u>VNr</u>, Zeit, Ort, Std.zahl, *PName*)
Übgsgruppe(<u>ÜNr</u>, Zeit, Ort, Std.zahl, *AName*, *VNr*)
Hiwi(<u>*Matrikel*</u> , *PName*, Vertrag,)
ÜbgsgruppeHiwi(<u>*ÜNr*</u>, <u>*Matrikel*</u>)
Student(<u>Matrikel</u>, Name, Adresse, Fach, Semester)

Ausführlich:



I. **Prof—hat—Assi** 1 Prof hat 0..mehrere Assis

1 Assi hat genau 1 Prof als Chef

→ alle Assis sind obligatorisch an der hat-Beziehung beteiligt (1-Seite) und damit mit ihr zusammen zu fügen

also

Prof(<u>PName</u>, …)

Assi(<u>AName</u>, …, <u>PName</u>) !Fremdschlüssel für die Verbindung NICHT VERGESSEN!

II. **Prof—hält—Vorlesung**

Zu jeder Vorlesung gehört ein Prof → Vorlesung ist obligatorisch an der Beziehungsrelation 'hält' beteiligt → verschmelzen

also

Prof(unverändert)

Vorlesung(<u>VNr</u>, …, <u>PName</u>) !Fremdschlüssel NICHT VERGESSEN!

III. **Assi—leitet--Übungsgruppen**

Jede Ü-Gruppe wird von genau einem Assi betreut → Ü-Gruppe ist obligatorisch an der Beziehungsrelation 'betreut' beteiligt → verschmelzen

also

Assi(unverändert)

Übungsgruppe(<u>ÜNr</u>, …, AName)

IV. **Übungsgruppen—gehört zu--Vorlesung**

1 Ü-Gruppe gehört zu genau einer Vorlesung → obligatorisch zu 'gehört-zu' → verschmelzen

also

Übungsgruppe(ÜNr, …, AName, VNr)

Vorlesung(unverändert)

V. **Student—IS A--Hiwi**

Ein Hiwi 'ist ein' Student , hat also alle Attribute des Studenten + seine zusätzlichen (speziellen). Hiwi ist der 'speziellere Entitätstyp' → Aufnahme des PS des allgemeinen Entitätstypen in sein Relationsschema

also

Student(Matrikel, …)

Hiwi(Matrikel, PName, Vertrag) PName = PS des Professors

VI. **Hiwi—korrigiert--Übungsgruppe**

n:m → nicht in weniger als 3 Tabellen erfassbar

also

Hiwi(Matrikel, PName, Vertrag)

Übungsgruppe(ÜNr, …, AName, VNr)

ÜbungsgruppeHiwi(Matrikel, ÜNr) die Beziehungsrelation 'korrigiert'

Aufgabe 3

Setzen Sie das ER-Diagramm Schule vollständig in ein relationales Modell um, indem Sie alle benötigten Tabellen mit ihren Köpfen angeben, die Primärschlüssel kennzeichnen und durch Pfeile zwischen den Attributen verschiedener Tabellen die Beziehungen verdeutlichen. Im Text wurden schon einige Tabellen eingeführt.

Fremdschlüssel sind kursiv gesetzt

- Lehrer (Lehrer-Nr, Name, Vorname, Titel)
- Lehrerfach (*Lehrer-Nr*, Fach)
- Schüler(Schüler-Nr, Name, Vorname, Geburtsdatum, Schuleintrittsdatum, PLZ, Wohnort, Geschlecht, StraßeNr, *LehrerNr*, Tutorgruppenbezeichnung)
- VorherBesSchulen(*Schüler-Nr*, Schule)
- Besucht(*Schüler-Nr*, *Kurs-Nr*, Note, Fehlstunden)
- Kurs (Kurs-Nr, Thema, Jahrgangsstufe, *Lehrer-Nr*, Fach, Typ)
- KursInRaum(*Kurs-Nr*, *Raum-Nr*, Zeit)
- Raum(Raum-Nr, Sitzplätze)
- Fachraum(Fach, *Raum-Nr*)
- FachraumAusstattung(*Raum-Nr*, Sonderausstattung)

Ausführlich:

I. **Lehrer—ist Tutor--Lehrer**

jeder Schüler hat 1 Tutor → obligatorisches Element von 'ist Tutor' → verschmelzen

also

Schüler(Schüler-Nr, Name, Vorname, Geburtsdatum, Schuleintrittsdatum, PLZ, Wohnort, Geschlecht, StraßeNr, *LehrerNr*, Tutorgruppenbezeichnung)

LehrerNr: Fremdschlüssel

Tutorgruppenbezeichnung für die Beziehung

Mehrfachattribut 'vorher besuchte Schule' → eigene Tabelle mit PS aus Schüler

aber zusammengesetzter PS nötig, da >1 vorherige Schule pro Schüler

also VorherBesuchteSchule(Schüler-Nr, Schule)

Lehrer(Lehrer-Nr., Titel, Name, Vorname)

Mehrfachattribut 'LehrerFächer' → eigene Tabelle mit PS aus Lehrer

aber zusammengesetzter PS nötig, da >1 Fach pro Lehrer-Nr

also LehrerFächer(Lehrer-Nr., Fach)

II. **Lehrer—besucht--Kurs**

n:m → 3 Tabellen

Schüler(unverändert)

besucht(Schüler-Nr, Kurs-Nr, Note, Fehlstunden)

Schüler-Nr: PS(E1)

Kurs-Nr PS(E2)

Kurs(Kurs-Nr, Thema, Jahrgangsstufe, Fach, Typ)

III. **Lehrer—hält--Kurs**

1 Kurs wird immer von 1 Lehrer gehalten → 'Kurs' ist obligatorisches Element von 'hält'

Lehrer(unverändert)

Kurs(Kurs-Nr., …, Lehrer-Nr)

IV. **Raum—is a--Fachraum**

In das Relationenschema des spezielleren Entity-Typs zusätzlich den PS des allgemeinen Typs aufnehmen, also

Fachraum: spezieller

Raum: allgemeiner → PS von ’Raum’ ins Relationenschema von ’Fachraum’ aufnehmen

Raum(Raum-Nr., Sitzplätze)

Fachraum(Raum-Nr, Fach)

SonderausstattungenFachraum(Raum-Nr, Sonderausstattung) → Mehrfachattribut

Zusammengesetzter PS, da >1 Sonderausstattung pro Raum

V. **Kurs—in--Raum**

Raum(unverändert)

Raum(unverändert)

KursImRaum(Raum-Nr, Kurs-Nr, Zeit)

1 Kurs in 1 Raum zu verschiedene Zeiten möglich

Aufgabe 4

a) Setzen Sie die unter “Modellierung von Ausschnitten der Wirklichkeit“ dargestellte Aufgabe der Firma Microtec in das relationale Modell um, indem Sie das ER-Modell nach den beschriebenen Verfahren in Tabellen umsetzen!

Artikel(Artikel-Nr, Bezeichnung, Verkaufspreis, Bestand)
Liefert(*Artikel-Nr*, *Hersteller-Nr*, Preis)
Hersteller(Hersteller-Nr, Name, Adresse)
Umfaßt(*Bestell-Nr*, *Artikel-Nr*, Stückzahl)
Bestellung(Bestell-Nr, Bestelldatum, *Kunden-Nr*)
Kunde(Kunden-Nr, Name, Adresse)

Aufgabe 5

Setzen Sie das ER-Modell der Spedition Hugo Unbedarft aus „Aufgaben zum ERM“ in ein relationales Modell um. Überlegen Sie dabei, welche Tabellen entstehen müssen und welche Beziehungen sich integrieren lassen.

Anmerkung:
Wenn Sie die Normalisierung erlernt haben kommen sie auf diese Aufgabe zurück und Überprüfen anschließend, inwieweit sich die entstandenen Tabellen in der 3. Normalform befinden! Treten Abhängigkeiten zwischen den Schlüsseln auf?

Auftrag(AuftragsNr, Bezeichnung, Ziel, Entfernung)
WirdDurchgeführt(AuftragsNr, LkwNr, Auftragsdatum)
Lkw(LkwNr, Typ, Maxzuladung, TÜV)
WirdGefahren(LkwNr, FahrerNr)
Fahrer(FahrerNr, Name, Vorname, Typ)

Die n:m-Beziehungen werden jeweils in 3 Tabellen abgebildet, diese befinden sich in der 3. NF nach Codd.

Aufgaben

Aufgaben zur Normalisierung

Aufgabe 1

Ein Hochschulberater berät Studenten, die alle im Wohnheim der Universität leben und alle das gleiche Hauptfach studieren. Aus Besprechungen mit Studenten will der Berater eine kleine Datenbank zur Unterstützung der Beratung entwickeln. Er legt folgende Attribute und Regeln fest:

SNr	Studentennummer, ganze Zahl , eindeutiger Schlüssel für Studenten
SName	Name des Studenten, nicht eindeutig
ZNr	Zimmernummer ganzzahlig; jeder Student hat ein Zimmer, das aber von mehreren Studenten bewohnt werden kann
TNr	Telefonnummer des Studenten; sie ist für alle Zimmerbewohner gleich
Kurs	Identifikationsnummer der Kurse, die ein Student belegt oder bereits abgeschlossen hat
Semest	Semester, in dem ein Kurs abgeschlossen wurde; ein Student darf den gleichen Kurs in einem späteren Semester wiederholen
Note	Note eines abgeschlossenen Kurses

Der Berater möchte folgende Daten in der Datenbank speichern (Auszug):

SNr	SName	ZNr	TNr	Kurs	Semest	Note
3215	Jonas Mike	120DH	2136	MAT122	W88	1,4
				PHY120	S88	2,5
				WIW330	W89	3,1
				MAT122	S89	1,2
3456	Schmidt Klaus	237VH	3127	MAT122	W87	3,2
				MAT130	S87	2,9
4576	Neider Paul	120DH	2136	PHY230	W88	2,8
				MAT120	S88	2,1

a) Erstellen Sie ein Diagramm zum ER-Modell für diese Daten.

b) Machen Sie aus der obigen Tabelle eine gültige Relation in der 1. Normalform. Welche Anomalien können dabei auftreten?

c) Erstellen Sie anhand des ER-Modells und der Umsetzungsregeln das relationale Modell. Begründen Sie, inwieweit die Tabellen die 3. Normalform erfüllen.

d) Geben Sie die relationalen Operationen an, um Kurslisten auszugeben.

Aufgabe 2

Ein kleines Belegkrankenhaus stellt auf EDV um. Die Patientendaten enthalten die folgenden Attribute in Tupelschreibweise:

PATIENT = (Name,Gebdatum, Adresse, Blutdruck, Datum, Diagnose,
Therapie, Datum, Diagnose, Therapie,...,Belegung)

Anmerkung: *Systol*: zusammenziehen; Kontraktion eines Hohlorgans

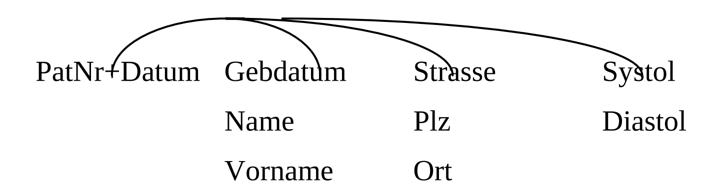
Diastol: erschlaffen; Erschlaffen eines muskulären Organs zwischen 2 Systolen

Beides bezieht sich auf den Blutdruck.

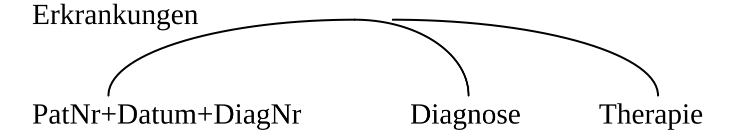
Erklären Sie anhand der folgenden Abbildungen den Gang der Normalisierung und begründen Sie die einzelnen Schritte. Bilden Sie dabei Klassen , gliedern Sie also die Attribute unter Oberbegriffen.

<u>1. Normalform</u>

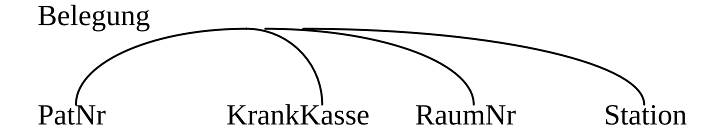
Patient



Erkrankungen



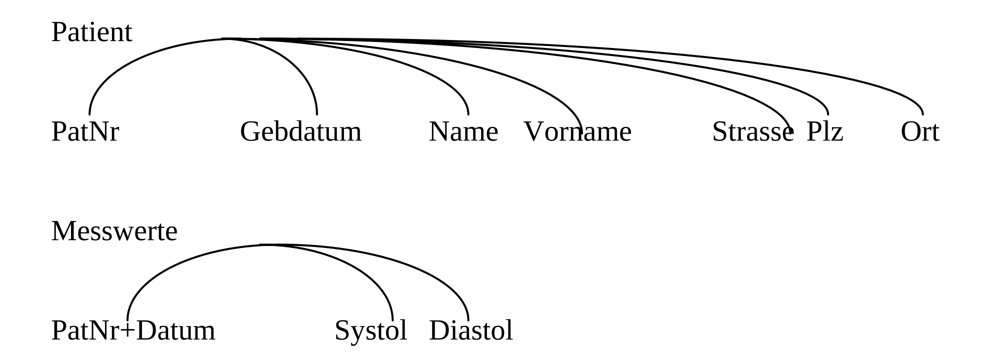
Belegung



<u>2. Normalform</u>

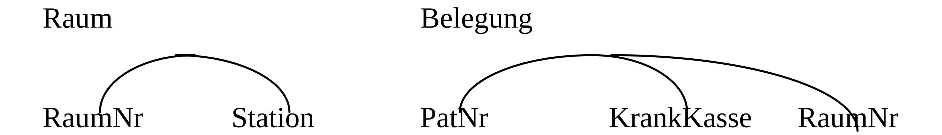
Die Datengruppe PATIENT wird weiter zerlegt:

Patient



3. Normalform

Die Datengruppe BELEGUNG wird weiter zerlegt:



Geben Sie für die 3. Normalform die entsprechenden Tabellen im relationalen Modell an!

Aufgabe 3

Sie haben die Aufgabe der Spedition von Hugo Unbedarf bereits gelöst und sich auch schon über die Tabellen und Verbindungen Gedanken gemacht, bevor Sie dieses Thema explizit erlernt haben. Schauen Sie sich jetzt die Aufgabe noch mal an und lösen Sie sie formal im Lichte Ihres jetzigen Kenntnisstands.

Aufgabe 4

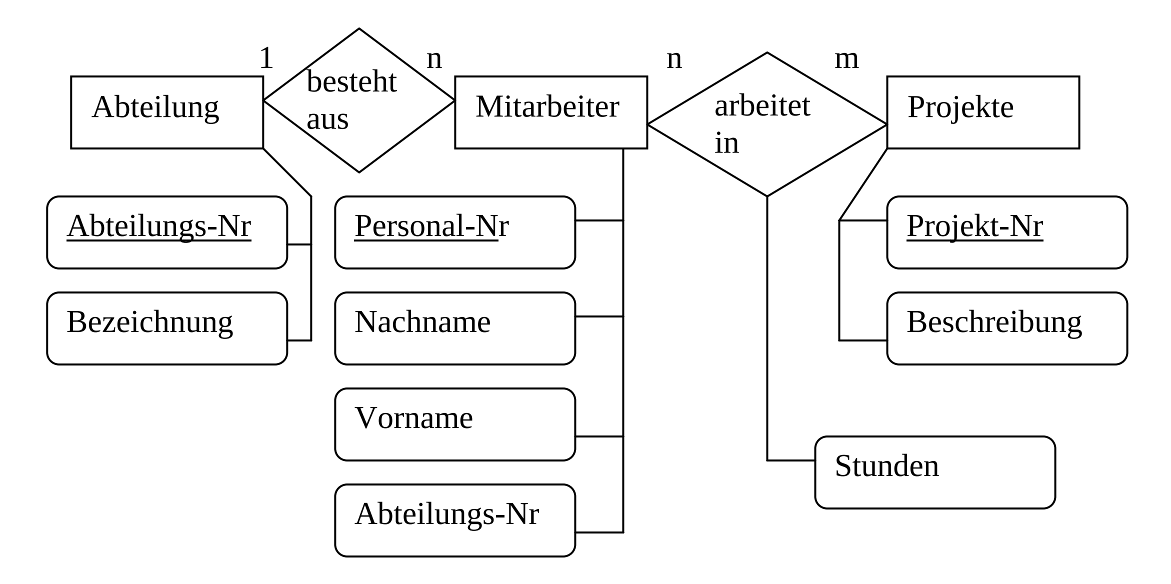
Eine Tabelle mit Lehrerdaten sei wie folgt formuliert

Nr	Nachname	Vorname	Amtsbez.	Besoldungsgruppe	Klasse
1	Bavaria	Eusebia	StRin	A13	11WA, 13ET
2	Bachmann	Hanna	StRin	A13	11WB
3	Kalkulus	Carl-Johann	StD	A15	12SC
4	Spike	Moses	OStR	A14	12GB

Analysieren Sie die Tabelle. Welche Schwächen fallen auf? Normalisieren Sie sie anschließend.

Aufgabe 5

Gegeben sei folgendes ERM.



Geben Sie die sich ergebende relationale Tabellenstruktur in der Kompaktschreibweise "Tabellenname(*PS*, Attribute)" an. Leiten Sie dabei Ihre Lösung ausführlich her und dokumentieren Sie die einzelnen Schritte.

Aufgabe 6

Sie importieren Daten aus einer Excel-Tabelle und erhalten die folgende Struktur. Normalisieren Sie sie bis zur 3. Normalform. Dokumentieren Sie dabei ausführlich Ihre Schritte mit der entsprechenden Begründung (Definition der Normalformen und ihre Folgerungen).

PersNr	Nachname	Vorname	Abteilung	Projekt	Stunden
1	Lorenz	Christian	Einkauf	Verkaufsanalyse	198, 201
2	Baumann	Peter	Verkauf	Optimierung	120, 189, 43
3	Petersen	Anna	Personal	Weiterbildung	120

Aufgabe 7

Reflektieren Sie nochmals über das systematische Vorgehen während des Normalisierungsprozesses. Dazu studieren Sie vorliegenden Text und ergänzen die sich ergebenden Tabellen.

Anmerkung: Es gibt mehrere denkbare Lösungen, also mehrere Tabellenstrukturen. Falls Ihre Idee nicht in die vorgesehenen Tabellen passt ergänzen Sie einfach die Spaltenanzahl, fragen sich aber kritisch, ob Ihre Lösung den gestellten Anforderungen genügt. Das primäre Ziel der Aufgabe ist die nochmalige intensive Reflexion des systematischen Normalisierungsprozesses.

Ein Bücherwurm Karlo möchte Ordnung in sein ganz privates Bücherchaos bringen, da ihm die Unmengen an über die Jahre erworbenen Büchern mittlerweile ein wenig über den Kopf gewachsen sind. Entsprechend entschließt er sich spontan eine Datenbank (DB) dafür zu

entwickeln. Seine Freundin studiert gerade fleißig Informatik und kommt am Wochenende vorbei, da möchte unser edler Gesell' natürlich ein wahres Kunstwerk mit allen Schikanen vorweisen können.

Als Regeln für gutes Datenbank-Design findet er folgende Weisheiten

- Tabellen sollten keine redundanten (sich wiederholende) Daten enthalten. Wenn in einer Tabelle immer wieder dasselbe eingegeben werden muss, stimmt etwas nicht.
- Bei der Konzeptentwicklung der DB (konzeptuelles Modell) bietet sich z.B. Excel an, da man damit leicht die Tabellen und ihre Attribute visualisieren kann, ohne auf die spezifischen Restriktionen eines echten DBMS achten zu müssen

Diese Regeln sind formalisiert als Normalisierung bekannt.

Als erster Schritt im DB-Design ist der Weltenausschnitt zu modellieren, den man gerade betrachtet. Bei einer Bücherdatenbank sind entsprechend die Attribute zu ermitteln, die in der DB verwaltet werden sollen. Es wären dies beispielsweise

- Titel (title)
- Verlag (publName)
- Veröffentlichungsjahr (year)
- Autoren (authorX)

Bei der Ermittlung dieser Attribute helfen Realweltdaten wie Karteikarten, Akten usw. Eine handschriftliche Bücherliste von Karlo hat beispielsweise folgende Struktur:

title	publName	year	author1	author2	author3
Linux	Addison-Wesley	2004	Meyer		
Web	Apress	2003	Schulz	Einstein	
Mathe	New Riders	2006	Schmitz		
Physik	Einstein-Verlag	1901	Newton	Schrödinger	Heisenberg
Lyrik	Federweiß	2006	Schulz		

Nun ja, also genau genommen hatte bei der Physik auch noch Max Planck seine Finger im Spiel, aber es war halt nur Platz für drei Autoren… Nicht sehr schön. Die Festlegung auf drei Autoren ist willkürlich, unflexibel und schlichtweg unschönes Design. Auch die Lösung alle Autoren in ein einziges Feld authors zu schreiben

authors
Newton, Schrödinger, Heisenberg, Planck

ist nicht so das wahre, da Karlo gehört hat man solle pro Attribut, also pro Spalte in einer Tabelle, nur *atomare Werte*, also ein Datum statt mehrerer, verwalten.
Nun denn, als Designkriterien für diese erste Hürde gelten folgende Weisheiten , die der *ersten Normalform*:

- Nur atomare Werte pro Spalte.
- Spalten mit gleichartigem Inhalt müssen entfernt werden.
- Für jede Gruppe zusammengehöriger Daten muss eine eigene Tabelle gebildet werden.
- Jeder Datensatz muss durch einen sog. Primärschlüssel eindeutig identifiziert werden.

Die dritte Regel ist schon erfüllt, da momentan in jeder Spalte nur Daten gesammelt werden, die auch zu einem Buch gehören. Regel 1 ist gleichfalls erfüllt, da das Problem der Mehrfachdaten pro Attribut von Karlo durch die drei Autorenspalten gelöst wurde, auch wenn sich dadurch Probleme mit Regel zwei eingefangen hat. Regel 4 ist leicht zu befolgen. Es wird einfach eine fortlaufende Nummer eingeführt (AutoWert), die *titleID*, und als Primärschlüssel verwendet. Diese Nummer identifiziert dann das Buch mit all seinen Attributen eindeutig. Entsprechend ergibt sich die Tabelle nun wie folgt:

Prima, alle Kriterien der ersten Normalform sind erfüllt! Selbst das Autorenanzahlproblem wurde gelöst. Allerdings mag es doch verdächtig anmuten alle Daten für n-Autoren n-mal vorrätig zu halten. Das kann nicht der Weisheit letzter Schluss sein.
Eine Tabelle wird in die *zweite Normalform* umgeformt, in dem

- Eine Tabelle in erster Normalform ist.
- Immer dann, wenn sich Inhalte in Spalten wiederholen, müssen die Tabellen in mehrere Teiltabellen zerlegt werden.
- Die Tabellen müssen durch sog. Fremdschlüssel miteinander verbunden werden.

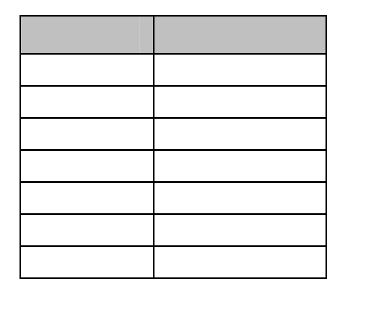
In obiger Tabelle wiederholen sich Daten auf Grund der Autoren. Eine Auslagerung derselbigen ergibt folgende Tabellenstruktur:

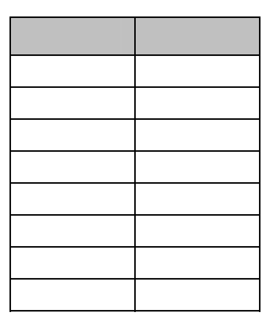
authID	titleID	author
1	1	Meyer
2	2	Schulz
3	3	Schmitz
4	4	Newton
5	4	Schrödinger
6	4	Heisenberg
7	4	Planck
8	5	Schulz

Schon besser. Allerdings kommt 'Schulz' als Autor immer noch mehrmals vor. Dies tritt immer dann auf, wenn ein Autor mehr als ein Buch geschrieben hat. Als Lösung käme eine weitere Aufteilung der Tabelle in Frage:

titles

authors *rel_title_author*





Letztere Tabelle muss von Rechts nach Links gelesen werden. Sie gibt also an, welche Bücher ein Autor – identifiziert über seine eindeutige authID – geschrieben hat. Vom Buch in einer Anfrage geht es also über die titleID zur authID, welche schließlich zu den Autoren des Buches führt. Die Tabelle rel_title_author hat übrigens einen *zusammengesetzten Primärschlüssel*, der aus den beiden Attributen besteht. Er identifiziert jedes Tupel eindeutig, daher ist ein zusätzlicher Primärschlüssel nicht nötig.
Es gibt noch eine weitere wichtige Normalform, die *dritte Normalform*. Diese löst indirekte (*transitive*) Abhängigkeiten auf. Ist A von B abhängig und B von C, so ist A transitiv von C abhängig, was nicht erlaubt ist, wenn eine Tabelle der dritten Normalform genügen soll. Ein typisches Beispiel ist die Postleitzahl und der Ort. In einer Mitarbeitertabelle werden die Personalstammdaten inklusive Wohnort erfasst. Eine MitarbeiterID (A) identifiziert den Mitarbeiter eindeutig, auch die zugehörige PLZ (B). Der Wohnort (C) ist aber abhängig von der PLZ (B) und damit besteht transitive Abhängigkeit zwischen MitarbeiterID und Wohnort. Entsprechend werden PLZ und Ort aus der Tabelle in eine eigene ausgegliedert.

PersID	Name	…	PLZ	Ort
1			33433	Marl
2			33453	Gelsenkirchen
3			2432	New York

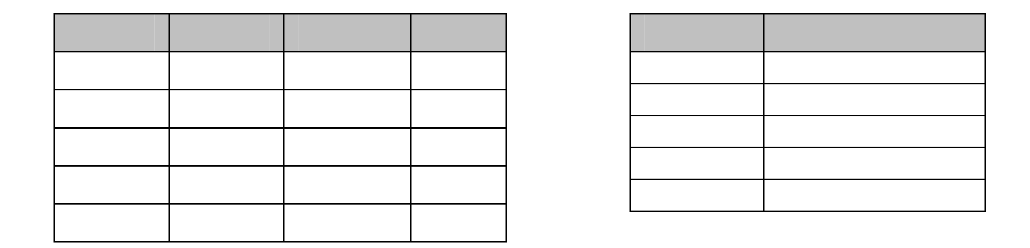
→

PersID	Name	…	PLZ
1			33433
2			33453
3			2432

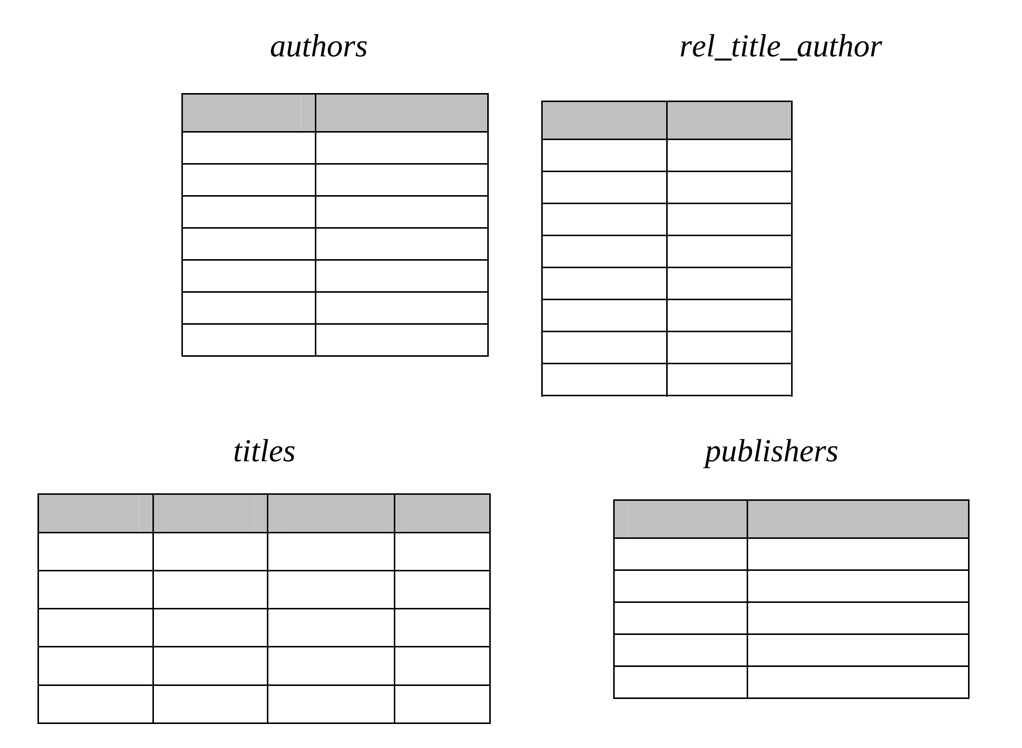
PLZ	Ort
33433	Marl
33453	Gelsenkirchen
2432	New York

Entsprechend reagiert Karlo und findet heraus, dass die Spalte *publName* in der *title*-Tabelle betroffen ist. Die Liste der Verlage und die Liste der Buchtitel sind voneinander abhängig und sollten daher getrennt behandelt werden. Es ist zwar nötig und sinnvoll zu jedem Buch den Verlag zu vermerken, aber es reicht doch ein kurzer Schlüssel statt immer der vollständigen Daten (hohen Fehleranfälligkeit und große Speicherplatzverschwendung, was zudem den Zugriff auf die Daten verlangsamt). Dies ist der Hintergrund für die dritte Normalform.

Gesagt, getan. Karlo macht folgendes daraus.



Die Tabellen *rel_title_author* und *authors* änders sich in der dritten Normalform nicht mehr. Die gesamte Bücherdatenbank besteht nun aus drei Tabellen.



Die Normalformen formalisieren demnach zwei Designprinzipien:

- Fasse in einer Tabelle nur das zusammen, was zusammen gehört.
- Vermeide Redundanzen und Abhängigkeiten durch Separierung in eine eigenen Tabelle.

Erweiterungen:

Die Datenbank könnte noch leicht um einige weitere – realistische – Features erweitert werden. So könnte die *titles*-Tabelle noch weitere Felder erhalten, wie etwa einen *Untertitel* oder einen *Kommentar*. Eine weitere languages-Tabelle könnte eine Liste aller Sprachen enthalten, in denen das Buch erschienen ist. Ein Verweis darauf könnte über ein *langID*-Feld in der Tabelle *titles* erfolgen. Eine Tabelle *categories* könnte eine Liste der Buchkategorien enthalten, die hierarchisch gegliedert ist, z.B. 'computer books → databases' und 'computer books → programming', 'computer books → programming → java' usw. Erfasst werden könnte diese Kategorie-Datailtabelle über ein Feld *catID* in der Tabelle *titles*.

Exkurs: Hierarchien

Categories ordnet Bücher in verschiedene Kategorien, die hierarchisch strukturiert sind. Dabei verweist *parentID* auf die übergeordnete Ebene.

- All Books
  - Children books
  - Computer books
    - Databases
      - Object oriented db
      - Relational db
      - Sql
    - Programming
      - Pearl
      - Php

→

catID	catName	parentID
1	computer books	11
2	databases	1
3	programming	1
…		
11	all books	11

Aufgabe 8

Eine Möbelfirma stellte vom Karteikartensystem

Artikelbezeichnung: ***Gabriele***		
Artikelart: ***Schreibtisch***	Artikelnummer: ***1000***	
Bestand:	Mindestbestand: ***5***	Höchstbestand: ***10***
Einkaufspreis:	Verkaufspreis:	
Lieferant: ***Tranel GmbH***	Name2: ***Büromöbel***	
Lieferer_Nr: ***102***	Anschrift: ***Bechemstr. 67, 47058 Duisburg***	

auf Excel um und möchte nun – mit steigendem Datenaufkommen – eine echte Datenbank nutzen. Es gibt sich nach dem Import folgende Tabellenstruktur. Normalisieren Sie sie.

Artikel_Nr	Artikelart	Artikel_Bez	LiefererNr	Name1	Name2	Anschrift
1000	Schreibtisch	Gabriele	102	Tranel GmbH	Büromöbel	Bechemstr., 33242 Duisburg
1001	Schreibtisch	Modern	101	Büromöbel AG	Büroeinrichtungen	Gutachtstr. 342, 12345 Bremen
1002	Schreibtisch	Exklusiv	102	Tranel GmbH	Büromöbel	Vogtweg 34, 32453 Verlin
1003	Büroschrank	Elegant	100	Wagner GmbH	Büromöbel	Becherstrasse 4, 34343 Oberhuisen
1004	Büroschrank	Aktuell	101	Büromöbel AG	Büroeinrichtungen	Gassengosse 2, 36999 Schwerten

Aufgabe 9

Ein Betrieb mit Außendienstmitarbeitern erfasst ihre Reisekosten in einer Tabelle, die wie folgt aufgebaut ist.

Reisekosten									
Rechnungs-nummer	Datum	Name	Vorname	Straße	PLZ	Ort	Kostenart	Anzahl	Einzelvergütung

Normalisieren Sie sie und dokumentieren Sie ausführlich Ihre jeweiligen Schritte mit Verweis auf die Fachbegriffe.

Überlegen Sie sich am Ende ferner, welche Verbindungen zwischen welchen Tabellen mit entsprechender Kardinalität auftreten könnten.

Aufgabe 10

Eine Tabelle sei in Kurzschreibweise wie folgt gegeben:

RECHNUNG(RechNr, KundenNr, Datum, Artikel)

Denken Sie sich exemplarische Datenbestände aus, die der 1. Normalform nicht genügen. Lösen Sie diese dann auf, bilden Sie also die 1. NF der Tabelle.

Aufgabe 11

Im Folgenden finden Sie nochmals eine kompakte Erklärung des Normalisierungsprozesses. Studieren Sie ihn sorgfältig.

Umsetzung des ERM in das relationale Datenbankmodell

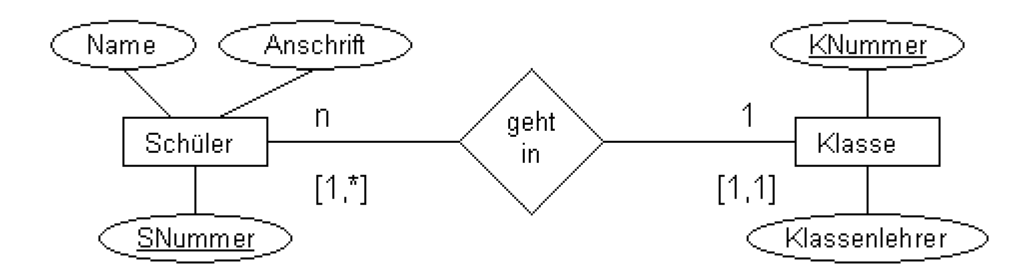
Grundregeln

1. **Entitätstypen:**
   - Jeder Entitätstyp wird in ein eigenes Relationsschema (Tabelle) abgebildet.
   - Schlüssel werden kenntlich gemacht.
2. **Beziehungstypen:**
   - Jeder Beziehungstyp wird in ein eigenes Relationsschema abgebildet.
   - Die Primärschlüssel der beiden beteiligten Entitätstypen werden zu zusätzliche Attributen des Relationsschemas (zus. zu evtl. vorhandenen eigenen Attributen).
   - Der Schlüssel des Relationsschemas bildet sich in Abhängigkeit vom Assoziationstyp wie folgt (Grobklassifikation):

Assoziationstyp	Schlüssel
1:1	einer der Primärschlüssel der beiden beteiligten Entitätstypen
1:n	der Primärschlüssel des zweiten Entitätstyps (also der "n-Entität")
n:m	beide Primärschlüssel der beteiligten Entitätstypen

Somit entstehen aus zwei mit einem Beziehungstyp verbundenen Entitätstypen immer 3 Tabellen.

Beispiel



wird zu

Relationsschema Schüler

SNummer	Name	Anschrift
0815	Schmidt	Mühleweg 7

Relationsschema geht_in

SNummer	KNummer
0815	8a

Relationsschema Klasse

KNummer	Klassenlehrer
8a	Herr Müller

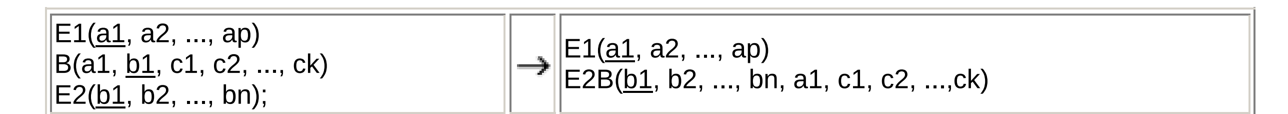
Optimierung

Die entstandenen 3 Tabellen lassen sich optimieren, falls **nicht** der Assoziationstyp n:m vorliegt.

Typ 1:n

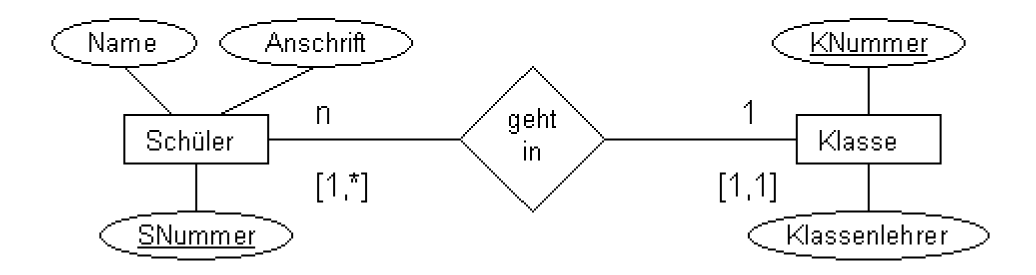
Es lassen sich die drei Relationsschemen u.U. auf zwei reduzieren.

Es gilt dann: Der Beziehungstyp mit all seinen Attribute und der "n-Entitätstyp" bilden eine neue Relation. Schlüssel wird das Schlüsselattribut des Entitätstyps der "n-Seite".



E1(a1, a2, ..., ap) B(a1, b1, c1, c2, ..., ck) E2(b1, b2, ..., bn);	→	E1(a1, a2, ..., ap) E2B(b1, b2, ..., bn, a1, c1, c2, ...,ck)

Beispiel



wird zu

Relationsschema Schüler

SNummer	Name	Anschrift	KNummer
0815	Schmidt	Mühleweg 7	8a

Relationsschema Klasse

KNummer	Klassenlehrer
8a	Herr Müller

Typ 1:c, c:c

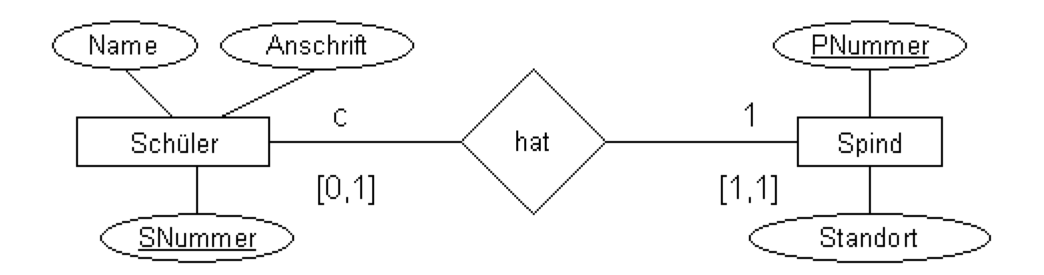
Es lassen sich die drei Relationsschemen auf zwei reduzieren.

Es gilt dann: Der Beziehungstyp mit all seinen Attribute und der "c-Entitätstyp" bilden eine neue Relation. Schlüssel wird das Schlüsselattribut des Entitätstyps der "c-Seite". Bei c:c kann der Fremdschlüssel NULL-Werte enthalten.



E1(a1, a2, ..., an) B(a1, b1, c1, c2, ..., ck) E2(b1, b2, ..., bm);	→	E1(a1, a2, ..., an) E2B(b1, b2, ..., bm, a1, c1, c2, ...,ck)

Beispiel



unter der Voraussetzung, dass jeder Schüler genau ein Spind hat, es aber auch nicht zugeordnete Spinde geben kann, wird zu

Relationsschema Schüler

SNummer	Name	Anschrift	PNummer
0815	Schmidt	Mühleweg 7	4563

Relationsschema Spind

PNummer	Standort
4563	Raum 218

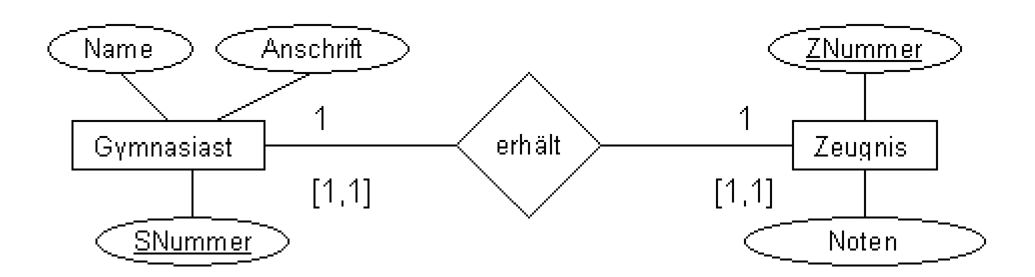
Typ 1:1

Es lassen sich die drei Relationsschemen auf eins reduzieren.

Es gilt dann: Der Beziehungstyp und die beiden Entitätstypen bilden eine neue Relation. Schlüssel wird ein Schlüsselattribut eines Entitätstyps.

E1(a1, a2, ..., an) B(a1, b1, c1, c2, ..., ck) E2(b1, b2, ..., bm);	→	E1BE2(a1, a2, ..., an,b1, b2, ..., bm, c1, c2, ...,ck)

Beispiel



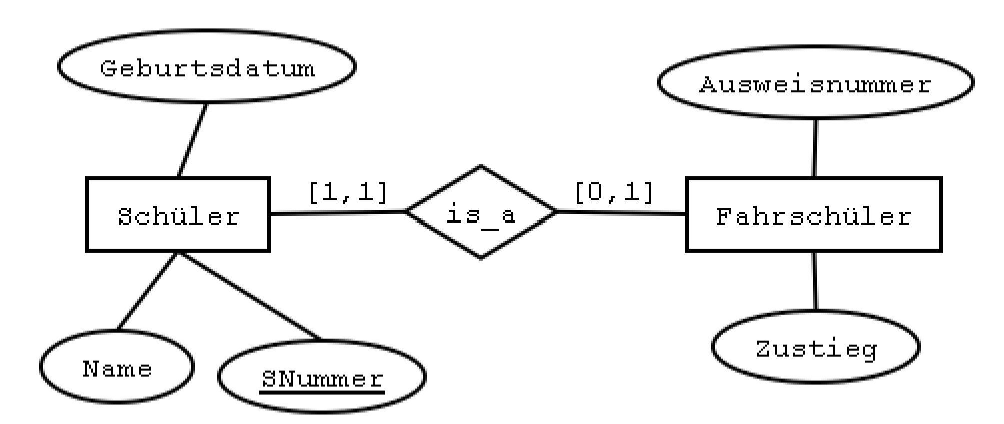
unter der Voraussetzung, dass jeder Gymnasiast auch ein (was auch immer für ein) Zeugnis erhält, wird zu

Relationsschema Gymnasiast_erhaelt_Zeugnis

SNummer	Name	Anschrift	Noten	ZNummer
0815	Schmidt	Mühleweg 7	1,2,...	4711

Sonderfälle

1. **is-a-Beziehungen**
   Is-a-Beziehungen drücken Spezialisierungen bzw. Generalisierungen aus. Dabei gilt, dass der Spezialentitätstyp alle Attribute des allgemeinen Entitätstyps erbt. Aus diesem Grund findet sich bei is_a-Beziehungen vom Typ 1:c an einem Entitätstyp (dem Speziellen) keine Schlüsselattributangabe. Is-a-Beziehungen lassen sich auflösen und in eine Tabelle integrieren. Hierzu muss nur der Primärschlüssel des allgemeineren Entitätstyps entweder als Fremdschlüssel (bei 1:mc) oder als Primärschlüssel (bei 1:c) in die Tabelle des speziellen Entitätstyps übernommen werden.



Relationsschema Schüler

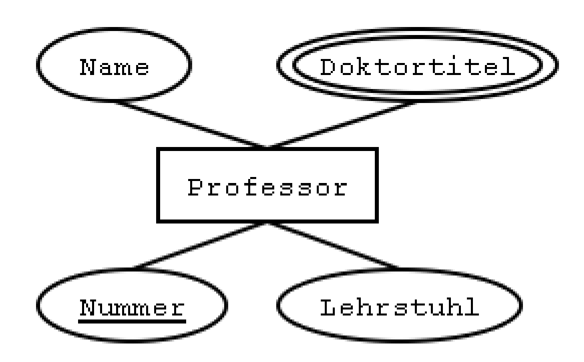
SNummer	Name	Geburtsdatum
0815	Schmidt	09.09.1990
4711	Paulsen	22.07.1987
007	Bond	02.05.1936

Relationsschema Fahrschüler

SNummer	Ausweisnummer	Zustieg
0815	8277728	Bahnhof
007	6525593	Hohenwulsch

2. **Mehrwertige Attribute**

Mehrwertige Attribute werden im relationalen Modell durch eine neue Tabelle dargestellt, die als Attribute das Primärschlüsselattribut des Entitätstyps und das Mehrfachattribut besitzen. Beide Attribute sind Schlüssel der Tabelle.



Relationsschema Professor

Nummer	Name	Lehrstuhl
0815	Schmidt	KI
4711	Paulsen	TI

Relationsschema Doktortitel

Nummer	Titel
0815	Dr. nat.
0815	Dr. phil.
4711	Dr. med. dent.

Lösungen

Aufgaben zur Normalisierung

Aufgabe 1

Die Objekte STUDENT, ZIMMER, KURS stehen über die Beziehungen WOHNT_IN und HAT_BELEGT miteinander in Beziehung. Die Beziehung WOHNT_IN hat die Komplexität 1:n, die Beziehung HAT_BELEGT die Komplexität m:n.

Die Wiederholungsgruppe KURS muß ausgegliedert werden, es entsteht eine neue Tabelle in der 1.NF. Zwischen ZNR und TNR besteht eine direkte Abhängigkeit, welche zu verschiedenen Anomalien führen kann.

Gliedert man die Tabelle mit den Attributen ZNR und TNR aus (1:1 -Beziehung) und integriert SEMESTER in HAT_BELEGT, so erfüllen die Tabellen die 3.NF, da keine Abhängigkeiten mehr vorhanden sind. Die Beziehung WOHNT_IN kann in die Tabelle STUDENT integriert werden, da es eine obligatorische Mitgliedschaft gibt. Da die Kursbezeichnung nicht eindeutig ist, kann entweder mit einem zusammengesetzten Schlüssel gearbeitet werden (KURS, SEMESTER) oder eine eindeutige Kursnummer vergeben werden.

Student(SNr, Sname, ZNr)
ZimmerTelefon(ZNr, TNr)
Kurs(KursNr, Kurs,Semester)
hat_belegt(SNr, KursNr, Note)

Die beteiligten Tabellen STUDENT, KURS und HAT_BELEGT werden über einen Equi-Join verbunden, da SNr bzw. KNr gemeinsame Schlüssel darstellen. Eine Selektion auf die betreffende Kursnummer liefert die Kursliste.

Aufgabe 2

PATIENT = (Name, Gebdatum, Adresse, Blutdruck, Datum, Diagnose,
Therapie, Datum, Diagnose, Therapie,...,Belegung)

Die Normalisierung bezieht sich im ersten Schritt auf die Zerlegung des Patiententupels in drei Tupel mit zusammengehörenden Attributen, praktisch den Klassen (aus der OOP) ’Patient, Erkrankungen, Belegung’.

PATIENT =

	PatNr	Name	Therapie1	Blutdruck	Belegung
	Datum	Gebdatum	Datum1		
		Adresse	Diagnose1		
			…		
Klasse	“Name“	“Patientendaten“	“Diagnose“		
Überklassen	Patientendaten, Diagnosedaten				

Nichtatomar: Adresse, Blutdruck, Name, Belegung

Wiederholungsgruppen: Therapie, Datum, Diagnose

→ es gibt nach der 1 NF kein … mehr

- Mehrfachattribute
- nichtatomare Werte

Nichtatomar:

Adresse: Strasse, PLZ, Ort

Blutdruck: Systol, Diastol

Name: Vorname, Nachname

Belegung: PatNr, KrankKasse, RaumNr, Station

Wiederholungsgruppen:

Therapie1 / Datum1 / Diagnose1 …:

Erkrankungen(PatNr, Datum, DiagNr, Diagnose, Therapie)

Im zweiten Schritt wird die Abhängigkeit vom gesamten PS gefordert. Es sind ja durch die 1 NF viele zusammengesetzte PS entstanden. In diesen Tabellen müssen jetzt die Bedingungen der Abhängigkeit vom gesamten PS geprüft werden.

Belegung: PS besteht nur aus 1 Attribut → automatisch in 2 NF

Erkrankungen: funktionale Abhängigkeit vom gesamten PS gegeben

Patient: PatNr identifiziert eindeutig

- GebDatum
- Name
- Vorname
- Strasse
- PLZ
- Ort

während

Systol

Diastol

vom gesamten PS abhängig sind, also

Messwerte PatNr+Datum

- Systol
- Diastol

Im dritten Schritt erfolgt die Auflösung indirekter (transitiver) Abhängigkeiten.

BELEGUNG: → RAUM(RaumNr, Station)

BELEGUNG(PatNr, Krankenkasse, RaumNr)

PATIENT: → PATIENT(PatNr, …, PLZ)

ORTE(PLZ, Ort)

Die Relationen lassen sich direkt aus der Grafik ablesen, da alle Tupel mit ihren Schlüsseln und Attributen bekannt sind und unerwünschte Abhängigkeiten beseitigt wurden.

Patient(PatNr, GebDatum, Name, Vorname, Strasse, Plz)

Orte(Plz, Ort)

Messwerte(PatNr, Datum, Systol, Diastol)

Raum(RaumNr, Station)

Belegung(PatNr, KrankKasse, RaumNr)

Erkrankungen(PatNr, Datum, DiagNr, Diagramme, Therapie)

Aufgabe 3

Sie haben die Aufgabe der Spedition von Hugo Unbedarf bereits gelöst und sich auch schon über die Tabellen und Verbindungen Gedanken gemacht, bevor Sie dieses Thema explizit erlernt haben. Schauen Sie sich jetzt die Aufgabe noch mal an und lösen Sie sie formal im Lichte Ihres jetzigen Kenntnisstands.

Auftrag(AuftragsNr, Bezeichnung, Ziel, Entfernung)
WirdDurchgeführt(AuftragsNr, LkwNr, Auftragsdatum)
Lkw(LkwNr, Typ, Maxzuladung, TÜV)
WirdGefahren(LkwNr, FahrerNr)
Fahrer(FahrerNr, Name, Vorname, Typ)

Die n:m-Beziehungen werden jeweils in 3 Tabellen abgebildet, diese befinden sich in der 3. NF nach Codd.

Aufgabe 4

Eine Tabelle mit Lehrerdaten sei wie folgt formuliert

Nr	**Nachname**	**Vorname**	**Amtsbez.**	**Besoldungsgruppe**	**Klasse**
1	Bavaria	Eusebia	StRin	A13	11WA, 13ET
2	Bachmann	Hanna	StRin	A13	11WB
3	Kalkulus	Carl-Johann	StD	A15	12SC
4	Spike	Moses	OStR	A14	12GB

Analysieren Sie die Tabelle. Welche Schwächen fallen auf? Normalisieren Sie sie anschließend.

Redundanzen: Ax taucht mehrmals auf (hier: A13)

→ Zusammenhang Amtsbez. ←→ Besoldungsruppen einmal in eigener Tabelle erfassen

Einfügeanomalie: Es kann keine neue Ax ohne einen Lehrer (Nr) geben, der sie hat (Situation z.B. bei DB-Neuanlage).

Löschanomalie: Ohne Mitarbeiter (alle gelöscht) sind auch die Amtsbez. und die Besoldungsgruppen weg.

Änderungsanomalie: Wird Ax geändert, so muss jeder einzelne Datensatz kontrolliert werden statt es einmal zentral zu formulieren durch Zuordnung Amtsbez. ←→ Besoldungsruppen

Transitive Abhängigkeit: Besoldungsgruppe ist nur von Amtsbez. Abhängig, nicht vom PS.

1. NF: Atomare Werte

nichtatomare Werte gibt es in der 1.Zeile

<u>Nr</u>	**Nachname**	**Vorname**	**Amtsbez.**	**Besoldungsgruppe**	**<u>Klasse</u>**
1	*Bavaria*	*Eusebia*	*StRin*	*A13*	*11WA*
1	*Bavaria*	*Eusebia*	*StRin*	*A13*	*13ET*
2	Bachmann	Hanna	StRin	A13	11WB
3	Kalkulus	Carl-Johann	StD	A15	12SC
4	Spike	Moses	OStR	A14	12GB

!!! Die Nr identifiziert jetzt NICHT mehr EINDEUTIG einen Datensatz (Zeile, Tupel). Daher ist der neue PS ein zusammengesetzter PS aus den Attributen *Nr* und *Klasse*. !!!

Sollte ein Lehrer in einer Klasse mehr als ein Fach unterrichten, so wäre der PS um das Attribut *Fach* zu ergänzen.

2. NF: Jedes Nicht-Schlüssel-Attribut ist funktional abhängig vom Gesamtschlüssel, nicht dagegen von Schlüsselteilen

Der PS ist zusammengesetzt, daher ist die 2.NF überhaupt noch zu prüfen.

Nachname, Vorname, Amtsbez. Und Besoldungsgruppe haben aber nichts mit den Klassen zu tun, die hängen nur von der Nr. ab. Ausgliedern in eine eigene Tabelle mit PS als Fremdschlüssel.

<u>Nr</u>	**Nachname**	**Vorname**	**Amtsbez.**	**Besoldungsgruppe**
1	Bavaria	Eusebia	StRin	A13
1	Bavaria	Eusebia	StRin	A13
2	Bachmann	Hanna	StRin	A13
3	Kalkulus	Carl-Johann	StD	A15
4	Spike	Moses	OStR	A14

<u>Nr</u>	**<u>Klasse</u>**
1	11WA
1	13ET
3	11WB
4	12SC
5	12GB

3.NF: Es gibt keine funktionalen Abhängigkeiten zwischen Nicht-Schlüssel-Attributen, es gibt also keine transitiven Abhängigkeiten.

Nr	**Nachname**	**Vorname**	**Amtsbez.**	**Besoldungsgruppe**

Nr	**Klasse**

Es gibt also transitive Abhängigkeiten. Auflösen durch eigene Tabelle

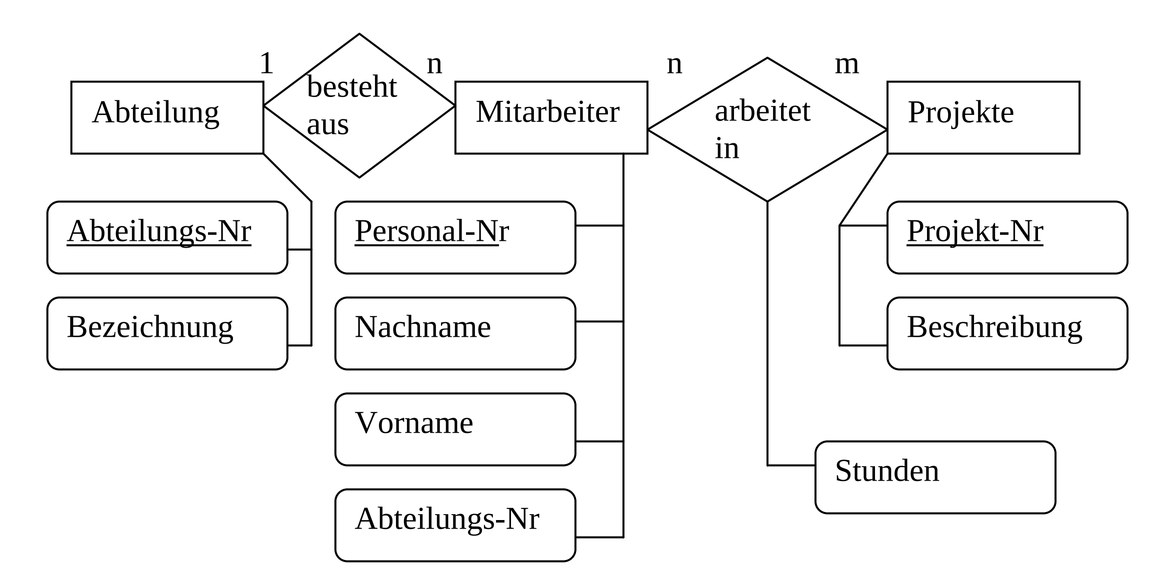
Nr	**Nachname**	**Vorname**	**Amtsbez.**
1	Bavaria	Eusebia	StRin
1	Bavaria	Eusebia	StRin
2	Bachmann	Hanna	StRin
3	Kalkulus	Carl-Johann	StD
4	Spike	Moses	OStR

Amtsbez	**Besoldungsgruppe**
StRin	A13
StD	A15
OStR	A14

Nr	**Klasse**
1	11WA
1	13ET
3	11WB
4	12SC
5	12GB

Aufgabe 5

Gegeben sei folgendes ERM.



Geben Sie die sich ergebende relationale Tabellenstruktur in der Kompaktschreibweise “Tabellenname(*PS*, Attribute)“ an. Leiten Sie dabei Ihre Lösung ausführlich her und dokumentieren Sie die einzelnen Schritte.

I. Abteilung—besteht aus--Mitarbeiter

Mitarbeiter ist obligatorisch an der Beziehungsrelation beteiligt → mit ihr vereinigen

ergibt

ABTEILUNG(<u>Abteilungs-Nr</u>, Bezeichnung)

MITARBEITER(<u>Personal-Nr</u>, Nachname, Vorname, Abteilungs-Nr) → ist nichts hinzu gekommen, da die Beziehungsrelation keine eigenen Attribute hat. Beziehung zur Abteilung wird über den Fremdschlüssel aufrecht erhalten.

II. Mitarbeiter—arbeitet in--Projekt

n:m → drei Tabellen

MITARBEITER(<u>Personal-Nr</u>, Nachname, Vorname, Abteilungs-Nr)

PROJEKT(<u>Projekt-Nr</u>, Beschreibung)

und die Beziehungsrelation mit den eigenen Attributen und den PS der beteiligten Entitäten als zusammengesetzter PS

MITARBEITERinProjekt(<u>Projekt-Nr</u>, <u>Personal-Nr</u>, Stunden))

Anmerkung:	Relationale Tabellenstrukturen, die aus einem ERM abgeleitet wurden, sind immer mindestens in der 2.Normalform.

Aufgabe 6

Sie importieren Daten aus einer Excel-Tabelle und erhalten die folgende Struktur. Normalisieren Sie sie bis zur 3. Normalform. Dokumentieren Sie dabei ausführlich Ihre Schritte mit der entsprechenden Begründung (Definition der Normalformen und ihre Folgerungen).

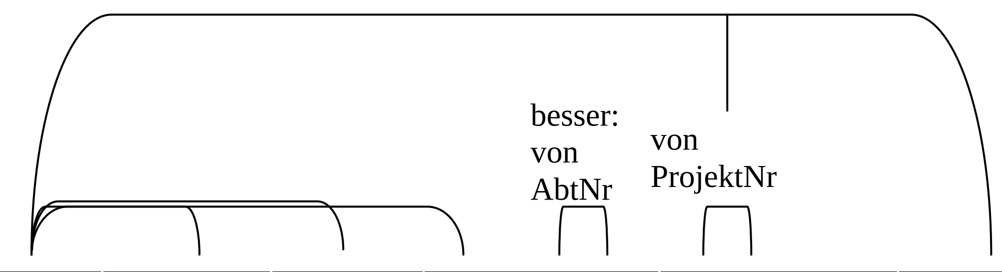
PersNr	Nachname	Vorname	Abteilung	Projekt	Stunden
1	Lorenz	Christian	Einkauf	Verkaufsanalyse	198, 201
2	Baumann	Peter	Verkauf	Optimierung	120, 189, 43
3	Petersen	Anna	Personal	Weiterbildung	120

1. Normalform nur atomare Werte erlaubt

PS muss einen Datensatz eindeutig identifizieren → i.d.R. sind dazu zusammengesetzte PS nötig

PersNr	Nachname	Vorname	Abteilung	Projekt	Stunden
1	Lorenz	Christian	Einkauf	Verkaufsanalyse	198
1	Lorenz	Christian	Einkauf	Verkaufsanalyse	201
2	Baumann	Peter	Verkauf	Optimierung	120
2	Baumann	Peter	Verkauf	Optimierung	189
2	Baumann	Peter	Verkauf	Optimierung	43
3	Petersen	Anna	Personal	Weiterbildung	120

Wir analysieren nun die Tabelle, um aus ihr die 2. und 3. NF ablesen zu können..



PersNr	Nachname	Vorname	Abteilung	Projekt	Stunden
1	Lorenz	Christian	Einkauf	Verkaufsanalyse	198
1	Lorenz	Christian	Einkauf	Verkaufsanalyse	201
2	Baumann	Peter	Verkauf	Optimierung	120
2	Baumann	Peter	Verkauf	Optimierung	189
2	Baumann	Peter	Verkauf	Optimierung	43
3	Petersen	Anna	Personal	Weiterbildung	120

2. Normalform Jedes Nicht-Schlüssel-Attribut ist funktional abhängig vom Gesamtschlüssel, nicht dagegen von Schlüsselteilen

Wir haben hier einen zusammengesetzten PS vorliegen, deshalb müssen wir uns überhaupt erst um die 2.NF kümmern, sonst wären die Bedingungen ja schon erfüllt.

Die Stunden hängen nur vom Projekt ab, nicht von der PersNr oder dem Nachnamen. Daher auslagern in eine eigene Tabelle. Gleiches gilt für den Abteilungsnamen.

Demnach

PersNr	Nachname	Vorname	AbteilungsNr	Abteilung
1	Lorenz	Christian	A	Einkauf
2	Baumann	Peter	B	Verkauf
3	Petersen	Anna	B	Personal

PERSONAL

Anmerkung: "AbteilungsNr" ist streng genommen für die Normalformen hier niicht ötig, jedoch dienen die Normalformen ja der Vermeidung von Inkonsistenzen und Redundanzen. Also führt man sie tunlichst ein und gliedert die beiden Objektbereiche "Personaldaten" und "Abteilung" in eigene Tabellen aus.

ProjektNr	Projekt
I	Verkaufsanalyse
II	Optimierung
III	Weiterbildung

PROJEKTE

ProjektNr	PersNr	Stunden
I	1	198
I	1	201
II	2	120
II	2	189
II	2	43
III	3	120

ARBEITENan

schafft Verbindungen zwischen den Tabellen

3. Normalform Es gibt keine funktionalen Abhängigkeiten zwischen Nicht-Schlüssel-Attributen, es gibt also keine transitiven Abhängigkeiten.

Die Abteilung ist nur von der "AbteilungsNr" abhängig, welche wiederum von der "PersNr" abhängig ist. Damit ist die "Abteilung" transitiv Abhängig von der "PersNr" und wird ausgegliedert.

PERSONAL(PersNr, Nachname, Vorname, AbteilungsNr)

→ AbteilungsNr für Verbindung nicht vergessen!

ABTEILUNG(AbteilungsNr, Abteilung)

PROJEKTE(ProjektNr, Projekt)

ARBEITETan(ProjektNr, PersNr, Stunden)

Aufgabe 7

Reflektieren Sie nochmals über das systematische Vorgehen während des Normalisierungsprozesses. Dazu studieren Sie vorliegenden Text und ergänzen die sich ergebenden Tabellen.

Anmerkung: Es gibt mehrere denkbare Lösungen, also mehrere Tabellenstrukturen. Falls Ihre Idee nicht in die vorgesehenen Tabellen passt ergänzen Sie einfach die Spaltenanzahl, fragen sich aber kritisch, ob Ihre Lösung den gestellten Anforderungen genügt. Das primäre Ziel der Aufgabe ist die nochmalige intensive Reflexion des systematischen Normalisierungsprozesses.

Ein Bücherwurm Karlo möchte Ordnung in sein ganz privates Bücherchaos bringen, da ihm die Unmengen an über die Jahre erworbenen Büchern mittlerweile ein wenig über den Kopf gewachsen sind. Entsprechend entschließt er sich spontan eine Datenbank (DB) dafür zu entwickeln. Seine Freundin studiert gerade fleißig Informatik und kommt am Wochenende vorbei, da möchte unser edler Gesell' natürlich ein wahres Kunstwerk mit allen Schikanen vorweisen können.

Als Regeln für gutes Datenbank-Design findet er folgende Weisheiten

- Tabellen sollten keine redundanten (sich wiederholende) Daten enthalten. Wenn in einer Tabelle immer wieder dasselbe eingegeben werden muss, stimmt etwas nicht.
- Bei der Konzeptentwicklung der DB (konzeptuelles Modell) bietet sich z.B. Excel an, da man damit leicht die Tabellen und ihre Attribute visualisieren kann, ohne auf die spezifischen Restriktionen eines echten DBMS achten zu müssen

Diese Regeln sind formalisiert als Normalisierung bekannt.

Als erster Schritt im DB-Design ist der Weltenausschnitt zu modellieren, den man gerade betrachtet. Bei einer Bücherdatenbank sind entsprechend die Attribute zu ermitteln, die in der DB verwaltet werden sollen. Es wären dies beispielsweise

- Titel (title)
- Verlag (publName)
- Veröffentlichungsjahr (year)
- Autoren (authorX)

Bei der Ermittlung dieser Attribute helfen Realweltdaten wie Karteikarten, Akten usw. Eine handschriftliche Bücherliste von Karlo hat beispielsweise folgende Struktur:

title	publName	year	author1	author2	author3
Linux	Addison-Wesley	2004	Meyer		
Web	Apress	2003	Schulz	Einstein	
Mathe	New Riders	2006	Schmitz		
Physik	Einstein-Verlag	1901	Newton	Schrödinger	Heisenberg
Lyrik	Federweiß	2006	Schulz		

Nun ja, also genau genommen hatte bei der Physik auch noch Max Planck seine Finger im Spiel, aber es war halt nur Platz für drei Autoren… Nicht sehr schön. Die Festlegung auf drei Autoren ist willkürlich, unflexibel und schlichtweg unschönes Design. Auch die Lösung alle Autoren in ein einziges Feld authors zu schreiben

authors
Newton, Schrödinger, Heisenberg, Planck

ist nicht so das wahre, da Karlo gehört hat man solle pro Attribut, also pro Spalte in einer Tabelle, nur *atomare Werte*, also ein Datum statt mehrerer, verwalten.
Nun denn, als Designkriterien für diese erste Hürde gelten folgende Weisheiten , die der *ersten Normalform*:

- Nur atomare Werte pro Spalte.
- Spalten mit gleichartigem Inhalt müssen entfernt werden.
- Für jede Gruppe zusammengehöriger Daten muss eine eigene Tabelle gebildet werden.
- Jeder Datensatz muss durch einen sog. Primärschlüssel eindeutig identifiziert werden.

Die dritte Regel ist schon erfüllt, da momentan in jeder Spalte nur Daten gesammelt werden, die auch zu einem Buch gehören. Regel 1 ist gleichfalls erfüllt, da das Problem der Mehrfachdaten pro Attribut von Karlo durch die drei Autorenspalten gelöst wurde, auch wenn sich dadurch Probleme mit Regel zwei eingefangen hat. Regel 4 ist leicht zu befolgen. Es wird einfach eine fortlaufende Nummer eingeführt (AutoWert), die *titleID*, und als Primärschlüssel verwendet. Diese Nummer identifiziert dann das Buch mit all seinen Attributen eindeutig. Entsprechend ergibt sich die Tabelle nun wie folgt:

titleID	title	publName	year	author
1	Linux	Addison-Wesley	2004	Meyer
2	Web	Apress	2003	Schulz
3	Mathe	New Riders	2006	Schmitz
4	Physik	Einstein-Verlag	1901	Newton
5	Physik	Einstein-Verlag	1901	Schrödinger
6	Physik	Einstein-Verlag	1901	Heisenberg
7	Physik	Einstein-Verlag	1901	Planck
8	Lyrik	Federweiß	2006	Schulz

Prima, alle Kriterien der ersten Normalform sind erfüllt! Selbst das Autorenanzahlproblem wurde gelöst. Allerdings mag es doch verdächtig anmuten alle Daten für n-Autoren n-mal vorrätig zu halten. Das kann nicht der Weisheit letzter Schluss sein.
Eine Tabelle wird in die *zweite Normalform* umgeformt, in dem

- Eine Tabelle in erster Normalform ist.
- Immer dann, wenn sich Inhalte in Spalten wiederholen, müssen die Tabellen in mehrere Teiltabellen zerlegt werden.
- Die Tabellen müssen durch sog. Fremdschlüssel miteinander verbunden werden.

In obiger Tabelle wiederholen sich Daten auf Grund der Autoren. Eine Auslagerung derselbigen ergibt folgende Tabellenstruktur:

titleID	title	publName	year
1	Linux	Addison-Wesley	2004
2	Web	Apress	2003
3	Mathe	New Riders	2006
4	Physik	Einstein-Verlag	1901
5	Lyrik	Federweiß	2006

authID	titleID	author
1	1	Meyer
2	2	Schulz
3	3	Schmitz
4	4	Newton
5	4	Schrödinger
6	4	Heisenberg
7	4	Planck
8	5	Schulz

Schon besser. Allerdings kommt 'Schulz' als Autor immer noch mehrmals vor. Dies tritt immer dann auf, wenn ein Autor mehr als ein Buch geschrieben hat. Als Lösung käme eine weitere Aufteilung der Tabelle in Frage:

titles

titleID	title	publName	year
1	Linux	Addison-Wesley	2004
2	Web	Apress	2003
3	Mathe	New Riders	2006
4	Physik	Einstein-Verlag	1901
5	Lyrik	Federweiß	2006

authors

authID	author
1	Meyer
2	Schulz
3	Schmitz
4	Newton
5	Schrödinger
6	Heisenberg
7	Planck

rel_title_author

titleID	**authID**
1	1
2	2
3	3
4	4
4	5
4	6
4	7
5	2

Letztere Tabelle muss von Rechts nach Links gelesen werden. Sie gibt also an, welche Bücher ein Autor – identifiziert über seine eindeutige authID – geschrieben hat. Vom Buch in einer Anfrage geht es also über die titleID zur authID, welche schließlich zu den Autoren des Buches führt. Die Tabelle rel_title_author hat übrigens einen *zusammengesetzten Primärschlüssel*, der aus den beiden Attributen besteht. Er identifiziert jedes Tupel eindeutig, daher ist ein zusätzlicher Primärschlüssel nicht nötig.
Es gibt noch eine weitere wichtige Normalform, die *dritte Normalform*. Diese löst indirekte (*transitive*) Abhängigkeiten auf. Ist A von B abhängig und B von C, so ist A transitiv von C abhängig, was nicht erlaubt ist, wenn eine Tabelle der dritten Normalform genügen soll. Ein typisches Beispiel ist die Postleitzahl und der Ort. In einer Mitarbeitertabelle werden die Personalstammdaten inklusive Wohnort erfasst. Eine MitarbeiterID (A) identifiziert den Mitarbeiter eindeutig, auch die zugehörige PLZ (B). Der Wohnort (C) ist aber abhängig von der PLZ (B) und damit besteht transitive Abhängigkeit zwischen MitarbeiterID und Wohnort. Entsprechend werden PLZ und Ort aus der Tabelle in eine eigene ausgegliedert.

PersID	Name	…	PLZ	Ort
1			33433	Marl
2			33453	Gelsenkirchen
3			2432	New York

→

PersID	Name	…	PLZ
1			33433
2			33453
3			2432

PLZ	Ort
33433	Marl
33453	Gelsenkirchen
2432	New York

Entsprechend reagiert Karlo und findet heraus, dass die Spalte *publName* in der *title*-Tabelle betroffen ist. Die Liste der Verlage und die Liste der Buchtitel sind voneinander abhängig und sollten daher getrennt behandelt werden. Es ist zwar nötig und sinnvoll zu jedem Buch den Verlag zu vermerken, aber es reicht doch ein kurzer Schlüssel statt immer der vollständigen Daten (hohen Fehleranfälligkeit und große Speicherplatzverschwendung, was zudem den Zugriff auf die Daten verlangsamt). Dies ist der Hintergrund für die dritte Normalform.
Gesagt, getan. Karlo macht folgendes daraus.

titleID	title	publID	year
1	Linux	1	2004
2	Web	2	2003
3	Mathe	3	2006
4	Physik	4	1901
5	Lyrik	5	2006

publID	publName
1	Addison-Wesley
2	Apress
3	New Riders
4	Einstein-Verlag
5	Federweiß

Die Tabellen *rel_title_author* und *authors* änders sich in der dritten Normalform nicht mehr. Die gesamte Bücherdatenbank besteht nun aus drei Tabellen.

authors

authID	author
1	Meyer
2	Schulz
3	Schmitz
4	Newton
5	Schrödinger
6	Heisenberg
7	Planck

rel_title_author

titleID	**authID**
1	1
2	2
3	3
4	4
4	5
4	6
4	7
5	2

titles

titleID	title	publID	year
1	Linux	1	2004
2	Web	2	2003
3	Mathe	3	2006
4	Physik	4	1901
5	Lyrik	5	2006

publishers

publID	publName
1	Addison-Wesley
2	Apress
3	New Riders
4	Einstein-Verlag
5	Federweiß

Die Normalformen formalisieren demnach zwei Designprinzipien:

- Fasse in einer Tabelle nur das zusammen, was zusammen gehört.
- Vermeide Redundanzen und Abhängigkeiten durch Separierung in eine eigenen Tabelle.

<u>Erweiterungen</u>:

Die Datenbank könnte noch leicht um einige weitere – realistische – Features erweitert werden. So könnte die *titles*-Tabelle noch weitere Felder erhalten, wie etwa einen *Untertitel* oder einen *Kommentar*. Eine weitere languages-Tabelle könnte eine Liste aller Sprachen enthalten, in denen das Buch erschienen ist. Ein Verweis darauf könnte über ein *langID*-Feld in der Tabelle *titles* erfolgen. Eine Tabelle *categories* könnte eine Liste der Buchkategorien enthalten, die hierarchisch gegliedert ist, z.B. 'computer books → databases' und 'computer books → programming', 'computer books → programming → java' usw. Erfasst werden könnte diese Kategorie-Datailtabelle über ein Feld *catID* in der Tabelle *titles*.

Exkurs: Hierarchien

Categories ordnet Bücher in verschiedene Kategorien, die hierarchisch strukturiert sind. Dabei verweist *parentID* auf die übergeordnete Ebene.

- All Books
  - Children books
  - Computer books
    - Databases
      - Object oriented db
      - Relational db
      - Sql
    - Programming
      - Pearl
      - Php

→

catID	catName	parentID
1	computer books	11
2	databases	1
3	programming	1
…		
11	all books	11

Aufgabe 8

Eine Möbelfirma stellte vom Karteikartensystem

Artikelbezeichnung: **Gabriele**		
Artikelart: **Schreibtisch**	Artikelnummer: **1000**	
Bestand:	Mindestbestand: **5**	Höchstbestand: **10**
Einkaufspreis:	Verkaufspreis:	
Lieferant: **Tranel GmbH**	Name2: **Büromöbel**	
Lieferer_Nr: **102**	Anschrift: **Bechemstr. 67, 47058 Duisburg**	

auf Excel um und möchte nun – mit steigendem Datenaufkommen – eine echte Datenbank nutzen. Es gibt sich nach dem Import folgende Tabellenstruktur. Normalisieren Sie sie.

Artikel_Nr	Artikelart	Artikel_Bez	LiefererNr	Name1	Name2	Anschrift
1000	Schreibtisch	Gabriele	102	Tranel GmbH	Büromöbel	Bechemstr., 33242 Duisburg
1001	Schreibtisch	Modern	101	Büromöbel AG	Büroeinrichtungen	Gutachtstr. 342, 12345 Bremen
1002	Schreibtisch	Exklusiv	102	Tranel GmbH	Büromöbel	Vogtweg 34, 32453 Verlin
1003	Büroschrank	Elegant	100	Wagner GmbH	Büromöbel	Becherstrasse 4, 34343 Oberhuisen
1004	Büroschrank	Aktuell	101	Büromöbel AG	Büroeinrichtungen	Gassengosse 2, 36999 Schwerten

0. Normalform

- nichtatomare Feldinhalte
- Lieferanten teils mehrfach erfasst → Mehrarbeit + Dateninkonsistenzgefahr

also

1. Normalform

- **Wiederholungsgruppen eliminieren**
- **Für atomare Werte sorgen**

Attributwert nicht atomar.

Anschrift
Bechemstr., 33242 Duisburg
Gutachtstr. 342, 12345 Bremen
Vogtweg 34, 32453 Verlin
Becherstrasse 4, 34343 Oberhuisen
Gassengosse 2, 36999 Schwerten

Das Datenfeld Anschrift ist daher in drei Tabellenfelder mit eindeutigen Informationen zu teilen.

Straße	PLZ	Ort
Bechemstr.	33242	Duisburg
Gutachtstr.	12345	Bremen
Vogtweg 34	32453	Verlin
Becherstrasse 4,	34343	Oberhuisen
Gassengosse 2	36999	Schwerten

Besser: für die Hausnummer ein eigenes Attribut vorsehen

Artikel_Nr	Artikelart	Artikel_Bez	LiefererNr	Name1	Name2	
1000	Schreibtisch	Gabriele	102	Tranel GmbH	Büromöbel	→
1001	Schreibtisch	Modern	101	Büromöbel AG	Büroeinrichtungen	
1002	Schreibtisch	Exklusiv	102	Tranel GmbH	Büromöbel	→
1003	Büroschrank	Elegant	100	Wagner GmbH	Büromöbel	
1004	Büroschrank	Aktuell	101	Büromöbel AG	Büroeinrichtungen	→

Straße	PLZ	Ort
Bechemstr.	33242	Duisburg
Gutachtstr.	12345	Bremen
Vogtweg 34	32453	Verlin
Becherstrasse 4,	34343	Oberhuisen
Gassengosse 2	36999	Schwerten

Keine Mehrfachattribute/Wiederholungsattribute (z.B. Liefer1,Datum1, Liefer2,Datum2, …) hier ok: ☺

Ein Artikel kann auch von mehreren Lieferanten geliefert werden. Daher ist ein zusammengesetzter PS nötig, um einen Datensatz eindeutig zu identifizieren. Wir wählen Artikel_Nr und LiefererNr.

2. Normalform

- **Attribute müssen vom gesamten PS abhängig sein, nicht schon von Teilend des PS**
- **Besteht der PS aus nur einem Attribut ist die 2 NF automatisch gegeben**

Es ergibt sich in der 2. NF die logische Aufteilung in Domänen. So werden Lieferantendaten in der Tabelle LIEFERANTEN gespeichert und Artikeldaten in der Tabelle LAGER.

In der Tabelle Lieferanten werden alle Lieferantendaten gespeichert und über den PS LiefererNr eindeutig identifiert

LiefererNr	Name1	Name2	PLZ	Ort
102	Tranel GmbH	Büromöbel	33242	Duisburg
101	Büromöbel AG	Büroeinrichtungen	12345	Bremen
102	Tranel GmbH	Büromöbel	32453	Verlin
100	Wagner GmbH	Büromöbel	34343	Oberhuisen
101	Büromöbel AG	Büroeinrichtungen	36999	Schwerten

In der Tabelle Lager werden alle Artikeldaten erfasst. Als Primärschlüssel wird die Spalte Artikel_Nr bestimmt. Alle anderen Spalten sind abhängig von dieser Spalte.

Artikel_Nr	Artikelart	Artikel_Bez	Bestand	Einkaufspreis
1000	Schreibtisch	Gabriele	45	60
1001	Schreibtisch	Modern	2	65
1002	Schreibtisch	Exklusiv	0	120
1003	Büroschrank	Elegant	19	100
1004	Büroschrank	Aktuell	24	250

(Bestand, Einkaufspreis nur zur Verdeutlichung weiterer sinnvoller Attribute mit aufgeführt. Kommen ja in der Ausgangstabelle gar nicht vor)

3 Normalform

- **tansitive (indirekte) Abhängigkeiten auflösen**

Die Tabelle ARTIKEL erfüllt das Kriterium schon. Die LIEFERANTEN Tabelle weist allerdings den Klassiker transitiver Abhängigkeit auf, die Abhängigkeit des *Ortes* von der *PLZ* und nicht vom PS L*ief*ererNr. Entsprechend entstehen zwei Tabellen.

LiefererNr	Name1	Name2	PLZ
102	Tranel GmbH	Büromöbel	33242
101	Büromöbel AG	Büroeinrichtungen	12345
102	Tranel GmbH	Büromöbel	32453
100	Wagner GmbH	Büromöbel	34343
101	Büromöbel AG	Büroeinrichtungen	36999

PLZ	Ort
33242	Duisburg
12345	Bremen
32453	Verlin
34343	Oberhuisen
36999	Schwerten

Insgesamt erhalten wir also folgendes Ergebnis:

LiefererNr	Name1	Name2	PLZ
102	Tranel GmbH	Büromöbel	33242
101	Büromöbel AG	Büroeinrichtungen	12345
102	Tranel GmbH	Büromöbel	32453
100	Wagner GmbH	Büromöbel	34343
101	Büromöbel AG	Büroeinrichtungen	36999

PLZ	Ort
33242	Duisburg
12345	Bremen
32453	Verlin
34343	Oberhuisen
36999	Schwerten

Artikel_Nr	Artikelart	Artikel_Bez	Bestand	Einkaufspreis
1000	Schreibtisch	Gabriele	45	60
1001	Schreibtisch	Modern	2	65
1002	Schreibtisch	Exklusiv	0	120
1003	Büroschrank	Elegant	19	100
1004	Büroschrank	Aktuell	24	250

Aufgabe 9

Ein Betrieb mit Außendienstmitarbeitern erfasst ihre Reisekosten in einer Tabelle, die wie folgt aufgebaut ist.

Reisekosten									
Rechnungs-nummer	Datum	Name	Vorname	Straße	PLZ	Ort	Kostenart	Anzahl	Einzelvergütung

Normalisieren Sie sie und dokumentieren Sie ausführlich Ihre jeweiligen Schritte mit Verweis auf die Fachbegriffe.

Überlegen Sie sich am Ende ferner, welche Verbindungen zwischen welchen Tabellen mit entsprechender Kardinalität auftreten könnten.

1. Normalform

Ok, alle Attribute sind schon atomar und Mehrfachattribute/Wiederholungsattribute gibt es nicht. Also keine Änderungen nötig.

2. Normalform

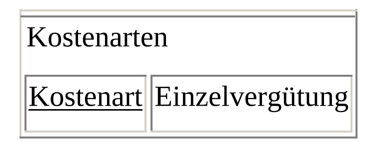
Es gibt hier einen zusammengesetzten PS, also ist die 2.NF überhaupt ein Thema. Auch hier liefert das formale Vorgehen wieder das logische Ergebnis, welches man auch erhalten hätte, wenn man „im Kopf“ die Attribute nach Zusammengehörigkeit geordnet hätte. Es ist volle funktionale Abhängigkeit vom GESAMTprimärschlüssel gefordert, also müssen wir die Tabellen aufteilen.

Wir stellen fest, dass die *Rechnungsnummer* viele Attribute eindeutig bezeichnet, nämlich *Datum*, *Name*, *Vorname*, *Straße*, *PLZ* und *Ort*.

Die *Einzelvergütung* wird bestimmt von der *Kostenart*, die *Anzahl* vom gesamten PS.

Also ergibt sich folgendes:

Reise						
Rechnungsnummer	Datum	Name	Vorname	Straße	PLZ	Ort



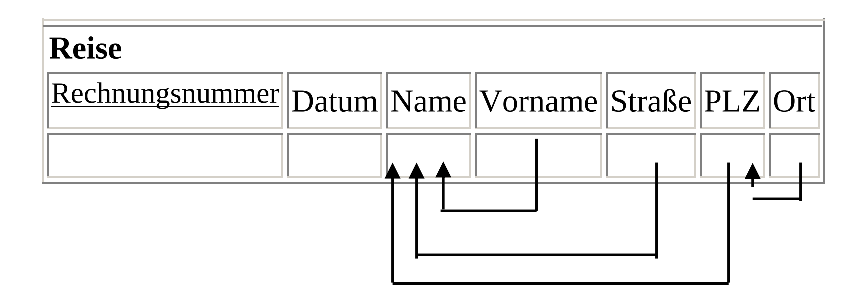
Kostenarten	
Kostenart	Einzelvergütung

Positionen		
Rechnungsnummer	Kostenart	Anzahl

3. Normalform

In der Tabelle "Reise" sind die Attribute *Vorname*, *Straße* und PLZ abhängig vom Attribut *Name*, nicht vom Primärschlüssel. Außerdem ist *Ort* abhängig von *PLZ*. Zu jeder *Rechnungsnummer* gehört eine *PLZ* und zu jeder *PLZ* ein *Ort*, also zu jeder *Rechnungsnummer* ein *Ort*.

Abhilfe:
Die transitiv abhängigen Datenfelder werden in weitere Tabellen ausgelagert, da sie nicht direkt vom Schlüsselkandidaten abhängen, sondern nur indirekt.
Da ein Name die Adresse bestimmt und nicht der momentane PS Rechnungsnummer, wird jedem Angestellten eine Personalnummer zugeordnet und diese Personaldaten auch in einer eigenen Tabelle erfasst. Diese ist Primärschlüssel der neuen Tabelle "Personal". Alternativ könnte ein zusammengesetzter Primärschlüssel aus Name, Vorname und Geburtsdatum benutzt werden (dieser sollte hinreichend eindeutig sein). Dieses Ergebnis erhielte man auch nach der ‚natürlichen" Aufteilung der Daten in Tabellen, denn die Rechnung hat ja nicht direkt etwas mit der Adresse zu tun, nur indirekt. Es handelt sich logisch betrachtet um unterschiedliche Domänen.



Reise						
Rechnungsnummer	Datum	Name	Vorname	Straße	PLZ	Ort

Personal				
Personalnummer	Name	Vorname	Straße	PLZ

Reise		
Rechnungsnummer	Datum	Personalnummer

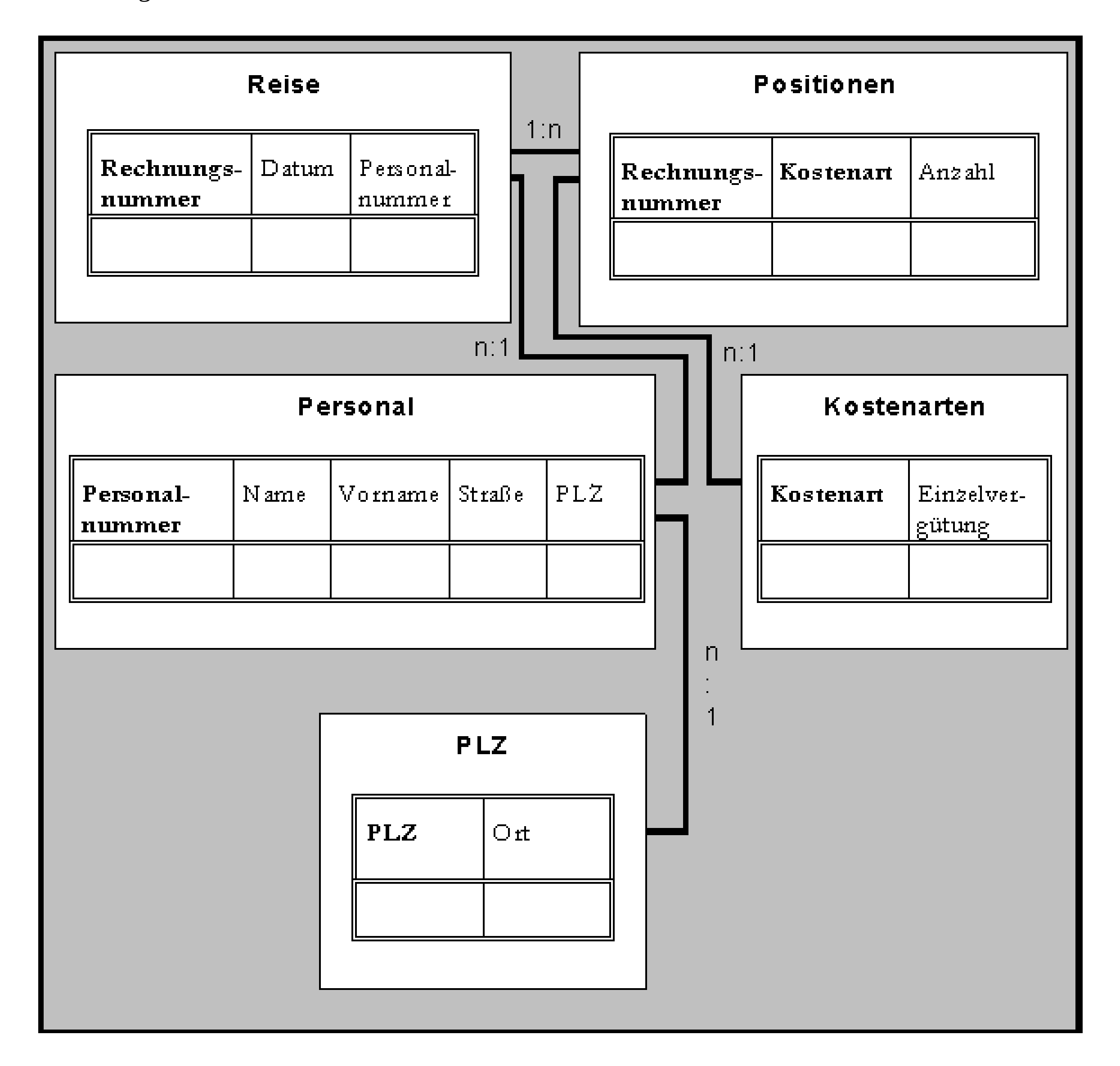
Kostenarten	
Kostenart	Einzelvergütung

Positionen		
Rechnungsnummer	Kostenart	Anzahl

PLZ	
PLZ	Ort

Wiederum besteht zwischen den Tabellen "Personal" und "Reise" sowie zwischen "PLZ" und "Personal" eine 1:n Beziehung, da ein Mitarbeiter mit einer bestimmten Personalnummer mehrere Dienstreisen machen kann und mehrere Mitarbeiter dieselbe Postleitzahl haben können.

Nach der dritten Normalisierung ergeben sich also folgende Tabellen, Datenfelder und Beziehungen:



Beachten Sie:

- Bei der Festlegung der Beziehungen kommt es auf die Reihenfolge der Tabellen an:
  Die Tabelle "PLZ" steht in einer 1:n Beziehung zur Tabelle "Personal", die Tabelle "Personal" dagegen in einer n:1 Beziehung (und nicht 1:n) zur Tabelle "PLZ"

- Eine Relationale Datenbank speichert nicht nur die Daten der Tabellen sondern auch die Beziehungen zwischen den Tabellen

Aufgabe 10

Eine Tabelle sei in Kurzschreibweise wie folgt gegeben:

RECHNUNG(RechNr, KundenNr, Datum, Artikel)

Denken Sie sich exemplarische Datenbestände aus, die der 1. Normalform nicht genügen. Lösen Sie diese dann auf, bilden Sie also die 1. NF der Tabelle.

RechNr	KundenNr	Datum	Artikel
1	2	03.03.2020	2 Apfelbäume
2	5	04.01.2008	2 Birnbäume, 3 Apfelbäume
3	12	01.01.1999	2 Pflaumenbäume, 3 Kirschbäume

Es sind nicht-atomare Attributwerte enthalten, also auflösen:

1. Normalform

RechNr	KundenNr	Position	Datum	Anzahl	Artikel
1	2	1	03.03.2020	2	Apfelbaum
2	5	1	04.01.2008	2	Birnbaum
2	5	2	04.01.2008	3	Apfelbaum
3	12	1	01.01.1999	2	Pflaumenbaum
3	12	2	01.01.1999	3	Kirschbaum

Nun ist ein zusammengesetzter PS nötig, um einen Datensatz eindeutig zu identifizieren. Dazu wurde das Feld Positon hinzugefügt, da ansonsten nur alle 4 Attribute zusammen einen eindeuten PS ergeben hätten.

Es gibt nun einen zusammengesetzten PS, daher ist die 2. NF überhaupt noch zu diskutieren.

Datum und *KundenNr* sind abhängig von der *Rechnungsnummer*, entsprechend muss ausgelagert werden.

2. Normalform

RECHNUNG

RechNr	KundenNr	Datum
1	2	03.03.2020
2	5	04.01.2008
2	5	04.01.2008
3	12	01.01.1999
3	12	01.01.1999

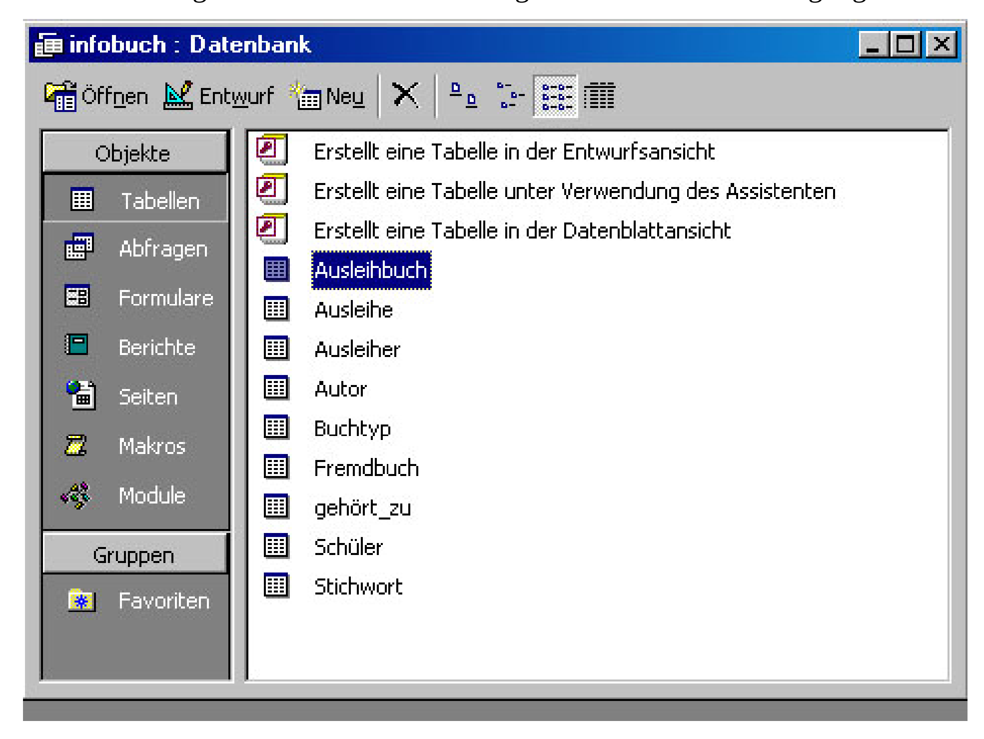
RECHNUNGDETAILS

RechNr	Position	Anzahl	Artikel
1	1	2	2 Apfelbäume
2	1	2	2 Birnbäume
2	2	3	3 Apfelbäume
3	1	2	2 Pflaumenbäume
3	2	3	3 Kirschbäume

4. Die Umsetzung des Modells mit Access

4.1 Tabellen und Beziehungen erstellen

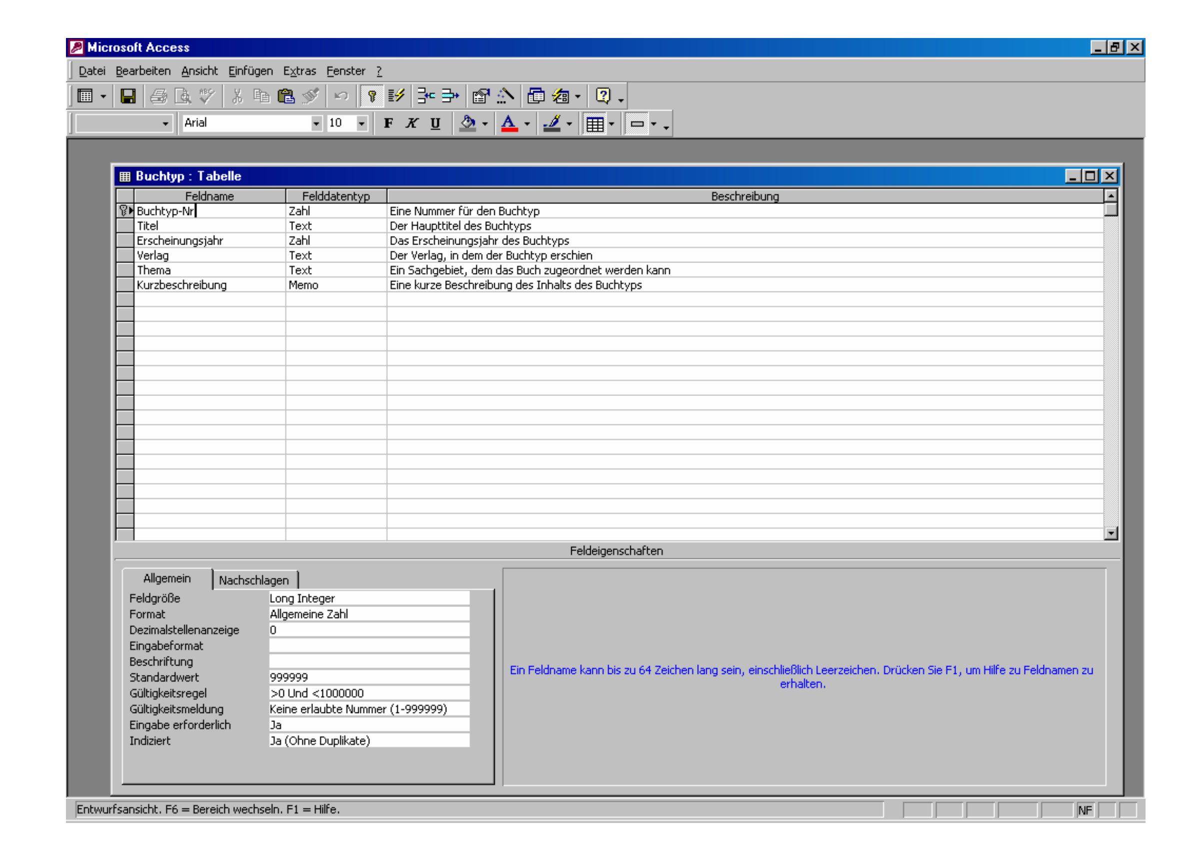
Wir wollen jetzt das erstellte relationale Modell konkret als Datenbank in *Access* realisieren. Es sei an dieser Stelle nochmals darauf hingewiesen, dass dieses Heft keinen Bedienungskurs für das konkrete Software-Paket enthält. Es wird vom Leser erwartet, dass er sich selbst mit dem Umgang mit dem Programm vertraut macht. Bedienerwissen wird nur soweit vermittelt, wie es für die Realisierung unserer konkreten Informatikbücherei unerlässlich ist. Für eine systematische Besprechung aller Menüs, Funktionen, usw. steht neben dem Originalhandbuch eine Unmenge an Literatur zur Verfügung



Hauptdatenbankfenster in Access

Wie bei praktisch allen Windows-Programmen enthält das Dateimenü die Punkte *Neue Datenbank* und *Vorhandene Datenbank öffnen*. Damit öffnet man das so genannte Datenbankfenster, in dem alle Teile der zu erstellenden Datenbank erfasst sind und das bei einer neuen Datenbank zunächst leer ist. Dabei interessieren wir uns zunächst nur für den Unterpunkt *Tabellen*. Wir müssen zunächst deren Struktur eingeben, was beim ersten Mal mit dem Schalter *Neu* geschieht, bei einer späteren Änderung der Struktur mit *Entwurf*. Die Datenbank-Software *Access* stellt für viele Aufgaben so genannte Assistenten zur Verfügung, das sind Hilfsprogrammen, die den Benutzer mit vorgefertigten Standardlösungen unterstützen, so auch beim Neuerstellen von Tabellen. Wir können hier aber getrost auf den elektronischen Helfer verzichten.

Es öffnet sich ein weiteres Fenster, in dem jetzt die Attributbezeichner, hier Feldnamen genannt, und ihre Wertebereiche (Felddatentyp gemeinsam mit Feldeigenschaften) eingegeben werden können. Die dritte Spalte *Bezeichnung* dient der genaueren Beschreibung des jeweiligen Attributs. Das untere Bild zeigt dieses Fenster für die Tabelle *Buchtyp*, wobei das Attribut *Buchtyp-Nr* gerade markiert ist.



Hier ist es wichtig, zunächst einmal die wichtigsten Datentypen aufzuzählen, die in *Access* möglich sind:

- **Text** — Beim Typ Text handelt es sich genauer gesagt um eine Zeichenkette der in *Feldgröße* angegebenen Maximallänge, höchstens jedoch 255 Zeichen. Dies ist der vermutlich am häufigsten benötigte Datentyp. Bei uns sind beispielsweise die Attribute *Titel*, *Autor*, *Vorname*, *Nachname*, aber auch *Inventar-Nr*, obwohl dafür bisher nur 12-stellige Zahlen benutzt wurden, vom Typ Text. Es soll die Möglichkeit offen bleiben, auch Inventarnummern der Form M34/156 zu verwalten, wie sie u.U. Abteilungen einer Schule benutzen.
- **Memo** — Memo wird für Texte mit großer Länge bis 64000 Zeichen verwendet. Felder dieses Typs können nicht indiziert werden, d.h. man kann keine einfache Sortierung bzw. alphabetische Suche vornehmen. Ein Beispiel für den Datentyp Memo ist bei uns das Attribut *Kurzbeschreibung* der Tabelle *Buchtyp*.
- **Zahl** — Damit lassen sich numerische Werte erfassen, was sich in *Feldgröße* noch präzisieren lässt. Es sind die Ganzzahltypen *byte*, *integer* und *long integer* möglich (vgl. diverse Programmiersprachen) und die Fließkommatypen *single* und *double*. Bezüglich der Genauigkeit und des erfaßten Bereichs dieser Zahlenarten schlage man im Handbuch oder ähnlichem nach. Bei uns sind die meisten Nummern wie *Buchtyp-Nr*, *Ausleihre-Nr*, usw. Ganzzahltypen.
- **Datum/Uhrzeit** — Hier können sowohl Datumswerte als auch Uhrzeiten eingegeben werden. Intern speichert das Programm eine Zahl und nur die im Eigenschaftsfeld *Format* gewählte Formatierung macht daraus ein Datum oder eine Uhrzeit in einer bestimmten Form. Bei uns taucht als Beispiel das Attribut *Ausleihdatum* auf.
- **Währung** — Hier können Währungsbeträge gespeichert werden. Der Typ unterscheidet sich von den Zahlen lediglich durch eine automatische

Formatierungsmöglichkeit in der voreingestellten Währung. Er taucht in unserem Beispiel nicht auf.

- **AutoWert** Hier erhöht das Programm bei jedem eingegebenen Datensatz den Feldinhalt um 1, d.h. man kann bei Attributen dieses Typs selbst keine Eingaben machen. Er eignet sich zur fortlaufenden Nummerierung von Datensätzen und wird bei uns für das Attribut *Stichwortnummer* verwendet. Werte des Zählers von Datensätzen, die gelöscht wurden, können nicht wieder verwendet werden.
- **Ja/Nein** entspricht dem *Boolean* von PASCAL. Erlaubte Eingaben sind lediglich *Ja* und *Nein* (auch *Wahr* und *Falsch* möglich). Kommt in unserem Beispiel nicht vor.
- **OLE-Objekt** dient z.B. dem Speichern von Grafiken oder Tönen aus anderen Programmen und soll wegen seines Spezialcharakters hier nicht weiter besprochen werden.

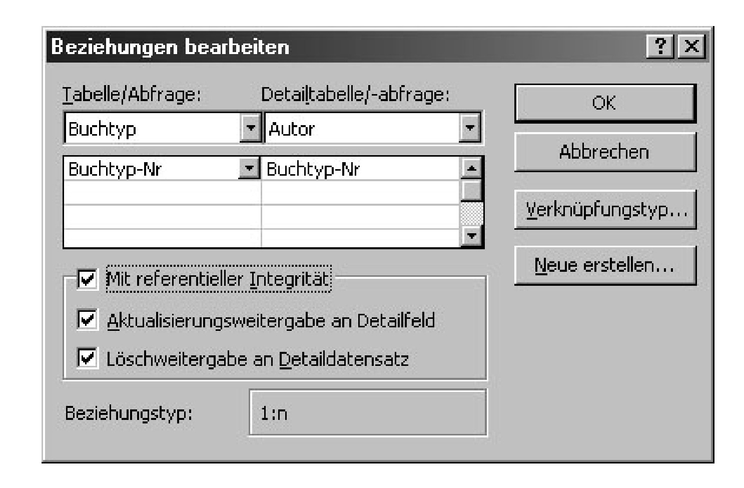
Hier tun sich viele Parallelen zu Datentypen in Programmiersprachen wie z.B. Java oder Delphi auf.

Mit Hilfe der Eingabe für "Feldeigenschaften" ist es möglich, weitere Präzisierungen für die später einzugebenden Werte vorzunehmen. So kann ein bestimmtes Format vorgeschrieben, es können Standardwerte gesetzt werden. Die Gültigkeitsregeln für den Wertebereich lassen sich genauer definieren und eine Fehlermeldung für die Überschreitung vereinbaren. Für Details sei auf die Handbücher bzw. die zahlreiche Literatur zu *Access* verwiesen. Wichtig ist noch das Einrichten eines **Primärschlüssels**. Das erreicht man am einfachsten durch Anklicken des Schlüsselsymbols, wobei das ausgewählte Feld markiert wird.

In ähnlicher Weise lassen sich auch die anderen Tabellen für unser Projekt entwerfen.

Nachdem nun alle Tabellen erstellt sind, müssen die Beziehungen zwischen ihnen hergestellt werden. Im Gegensatz zu unserem ER-Modell handelt es sich bei den von *Access* verwalteten Beziehungen um solche zwischen Datenfeldern verschiedener Tabellen. Dabei spielt es keine Rolle, ob die Tabelle ursprünglich einen Entity-Typ repräsentierte wie z.B. die Tabelle *Ausleihbuch* oder um eine Beziehungstabelle wie z.B. *Ausleihe.* Es kommen nur 1:1-Beziehungen zwischen Datenfeldern infrage wie z.B. zwischen *Ausleiher* und *Schüler* bezüglich des gemeinsamen Datenfeldes *Ausleiher-Nr* oder 1:n-Beziehungen wie z.B. zwischen *Ausleiher* und *Ausleihe* bezüglich desselben gemeinsamen Schlüssels. *Access* stellt die 1:n-Beziehung durch das Symbole 1 und ∞ an den Rechtecken für die Tabellen dar. *Die n:m-Beziehungen müssen in zwei 1:n-Beziehungen aufgelöst werden, was immer möglich ist.*

Das Definieren einer solchen Beziehung lässt sich in *Access* sehr einfach am Bildschirm durch „Ziehen" mit der Maus erreichen. Dazu klickt man unter dem Menüpunkt *Bearbeiten* den Unterpunkt *Beziehungen* an. Es erscheinen ein leeres Fenster und eine geänderte Menüleiste. Mit dem Stichwort *Beziehungen* lassen sich jetzt die Tabellen übernehmen, zwischen denen Beziehungen hergestellt werden sollen. Mit der Maus lässt sich nun eine 1:n - Beziehung zwischen zwei Feldern einer Tabelle herstellen, wobei man beim 1-Feld beginnt. Im angegebenen Beispiel wurde die Beziehung zwischen *Ausleiher-Nr* in der Tabelle *Ausleiher* und *Ausleiher-Nr* in der Tabelle *Ausleihe* hergestellt. Notwendig bei 1:n-Beziehungen ist es in *Access*, dass es sich beim 1-Feld um den Primärschlüssel dieser Tabelle handelt. Nach dem Ziehen öffnet sich das folgende Fenster:

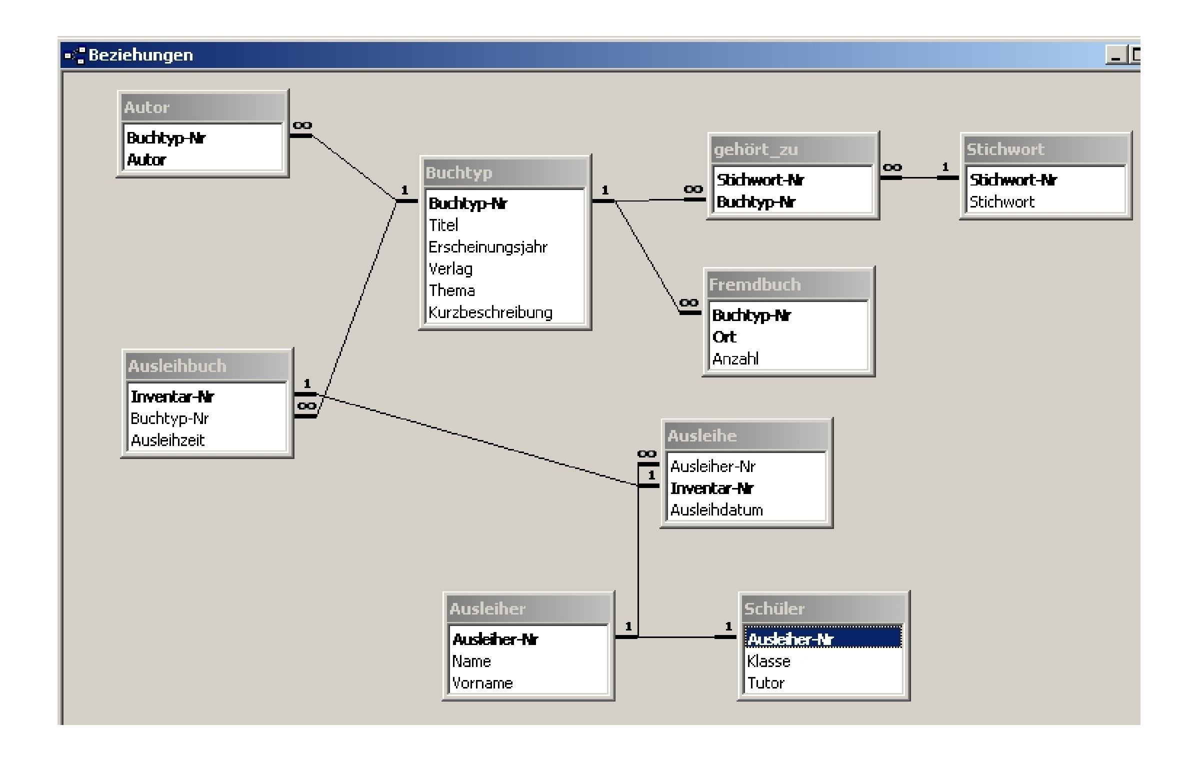


Im Fenster muss noch die so genannte referentielle Integrität bestätigt werden. Dadurch wird das System veranlasst, auf Datenkonsistenz zu achten. So ist es dann nicht möglich, ein Buch in der Tabelle *Ausleihe* an eine *Ausleiher-Nr* zu koppeln, die in der Tabelle *Ausleiher* nicht existiert. Änderungen der Nummer eines Ausleihers in *Ausleiher* werden an *Ausleihe* weitergegeben. Auch Löschoperationen werden weitergeleitet, d.h. mit dem Ausleiher würden auch alle Ausleihvorgänge gelöscht, die den jeweiligen Ausleiher betreffen. Das System weist jedoch auf das Löschen in verknüpften Tabellen hin und verlangt eine zusätzliche Bestätigung.

Bedenken Sie demnach bei jeder Verknüpfung, ob Sie referentielle Integrität sowie Lösch-/Aktualisierungsweitergabe wünschen. Die Optionen stellen Datenkonsistenz sicher, sind jedoch nicht für jede Verknüpfung sinnvoll.

Nach Erstellen aller Beziehungen ergibt sich das auf der nächsten Seite dargestellte Bild, das alle zu erstellenden Beziehungen umfasst und die Struktur unserer Datenbank repräsentiert. Damit sind wir so weit, dass wir Daten eingeben, verändern und löschen könnten. Dazu muss im Hauptfenster lediglich die entsprechende Tabelle geöffnet werden. Auf Einzelheiten der Dateneingabe soll hier nicht eingegangen werden, sie ist intuitiv leicht erlernbar.

Bedenke: **In der Praxis erfolgt die Eingabe der Daten jedoch nicht direkt auf den Tabellen, sondern über Formulare.** Warum sollte Ihnen mittlerweile klar sein. Wenn nicht, sprechen Sie Ihren Lehren drauf an.


Beziehungen
Autor
Buchtyp-Nr
Autor
Buchtyp
Buchtyp-Nr
Titel
Erscheinungsjahr
Verlag
Thema
Kurzbeschreibung
gehört_zu
Stichwort-Nr
Buchtyp-Nr
Stichwort
Stichwort-Nr
Stichwort
Fremdbuch
Buchtyp-Nr
Ort
Anzahl
Ausleihbuch
Inventar-Nr
Buchtyp-Nr
Ausleihzeit
Ausleihe
Ausleiher-Nr
Inventar-Nr
Ausleihdatum
Ausleiher
Ausleiher-Nr
Name
Vorname
Schüler
Ausleiher-Nr
Klasse
Tutor

Aufgaben



Aufgaben zum Erstellen von Tabellen

Aufgabe 1

Vergleichen Sie die Datentypen in einer Programmiersprache wie *Pascal* mit den Datentypen von *Access*. Wo gibt es Gemeinsamkeiten, wo Unterschiede, wo bestehen prinzipielle Unterschiede?

Aufgabe 2

Erstellen Sie die Tabellen für das Beispiel *Schulverwaltung* mit den entsprechenden Datentypen und Feldeigenschaften in *Access*.

Aufgabe 3

Erstellen Sie die Beziehungen für das Beispiel Schulverwaltung mit *Access!*

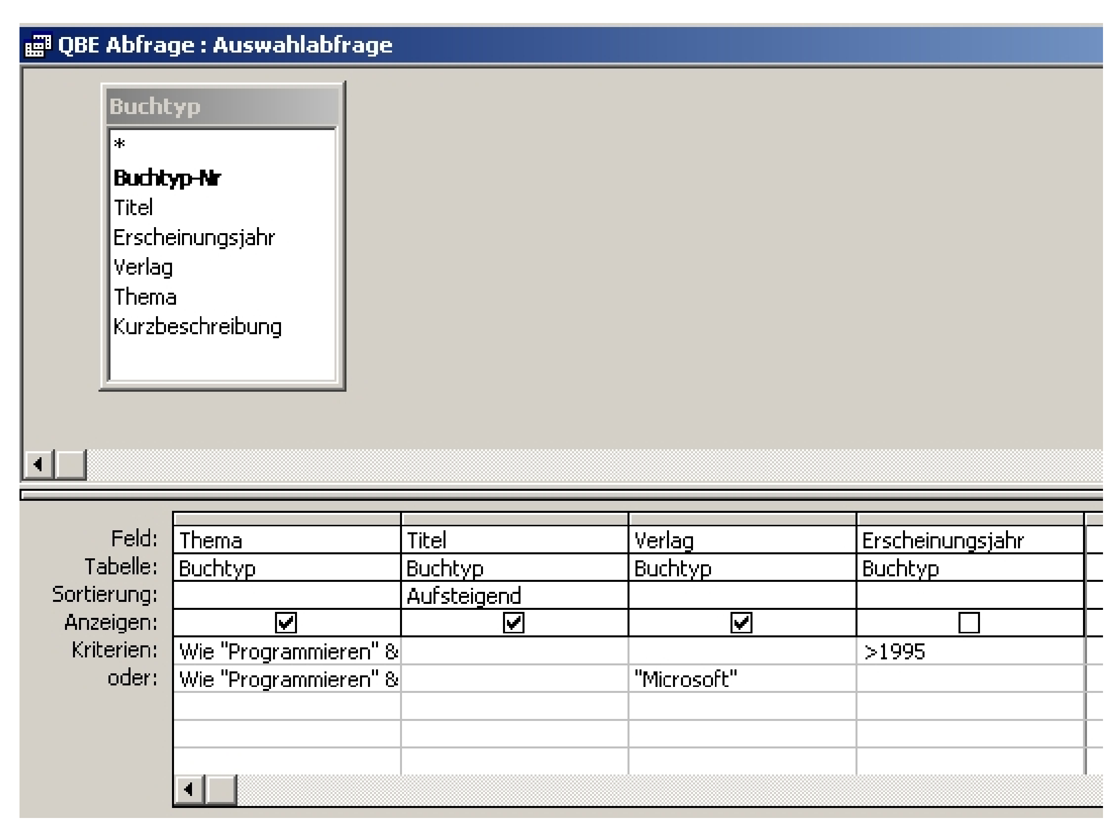
4.2 Abfragen in Access- mit QBE

Im letzen Kapitel wurde das Thema Abfragen schon etwas theoretischer als Anwendung relationaler Operatoren behandelt. Wir wollen nun an zwei Beispielen aufzeigen, wie Abfragen in *Access* erstellt und durchgeführt werden.

Um Abfragen in Access zu erstellen, kann man sich der QBE-Methode (Query by Example) bedienen. Dieses Verfahren lässt sich am einfachsten anhand unseres Projektes erläutern. Wir suchen nach allen Titeln, bei denen das Thema mit “Programmieren“ beginnt und die neuer als 1995 sind oder die bei “Microsoft“ verlegt wurden. Dazu öffnet man zunächst im Datenbankfenster eine neue Abfrage, analog dem Erstellen neuer Tabellen. Man muss nun ein oder mehrere Tabellen als Grundlage der Abfrage wählen. Die ausgewählten Tabellen erscheinen dann im Abfrageentwurffenster. Für unser erstes Beispiel reicht die Tabelle *Buchtyp* aus.

Im unteren Teil des Fensters können nun Felder der Tabelle gewählt werden, auf die sich die Abfrage bezieht, entweder durch Ziehen aus dem oberen Teil oder mit dem Auswahlpfeil rechts in der Feldzelle. So wählen wir nacheinander Thema, Titel, Verlag und Erscheinungsjahr aus. In der zweiten Zeile können Sortierungen eingestellt werden (aufsteigend oder absteigend). Bei mehreren Sortierungen werden die linken zuerst durchgeführt. In unserem Fall sollen die Titel alphabetisch dargestellt werden. Die dritte Zeile legt fest, ob das angegebene Feld im Ergebnis der Abfrage angezeigt wird. Hier wird auf die Anzeige des Erscheinungsjahres verzichtet, obwohl es beim Auswerten berücksichtigt wird.

Beachte: Anzeigen und Berechnen sind auch hier wieder zwei zu unterscheidende Ebenen!



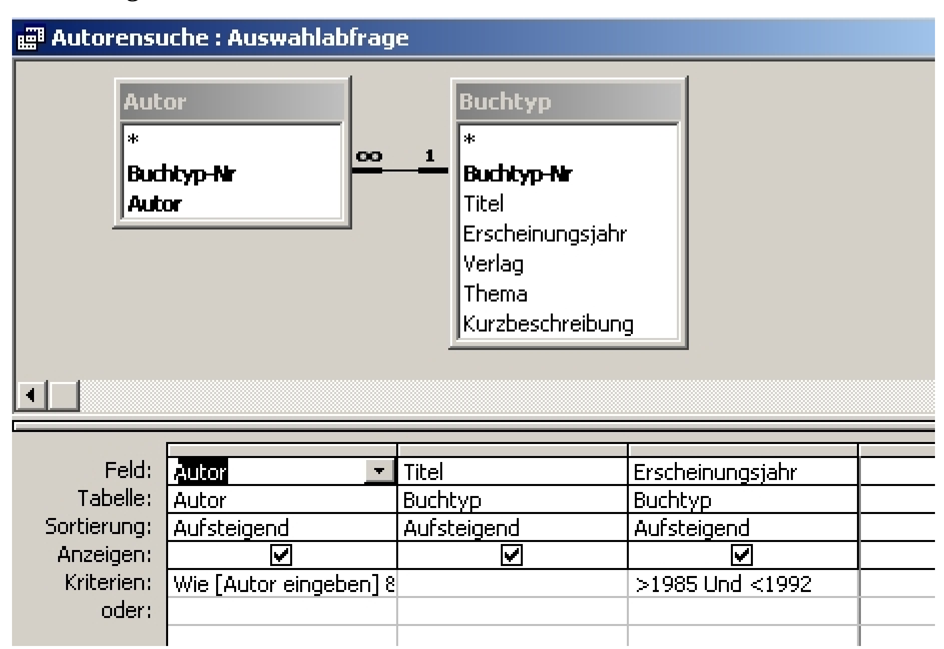
QBE-Abfrage

In den nächsten Zeilen können Kriterien für jedes Feld eingegeben werden, die vom einfachen Prüfen der Gleichheit wie bei „*Microsoft*“ über die Verwendung von Operatoren bei *>1995* bis hin zum Einsatz von Funktionen wie der WIE-Funktion reicht. Die WIE-Funktion gestattet vielfältige Variationen von Mustern, denen die zu findende Zeichenkette gehorchen soll. Bei uns bedeutet *WIE "Programmieren"&"*"* ein Wort, das mit „Programmieren“ beginnt und dann beliebig fortgesetzt wird.

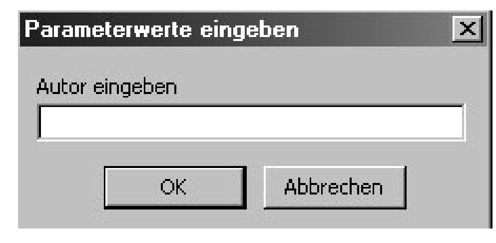
Kriterien in derselben Zeile sind UND-verknüpft, Kriterien untereinander mit ODER. Die erste Zeile liefert also alle Bücher, deren Titel mit „Programmieren“ beginnt und die neuer als 1995 sind. Die zweite Zeile liefert zusätzlich alle Bücher, die genauso beginnen, aber von „Microsoft“ verlegt werden. Das Erscheinungsjahr ist dabei ohne Bedeutung.

Dieses erste Beispiel zeigt, dass sich mit dieser Methode rasch und einfach Abfragen mit durchaus komplexen Kriterien gestalten lassen. Praktisch genau so einfach sind Abfragen über mehrere verknüpfte Tabellen.

Dazu wollen wir *Autor* und *Titel*, die ja bekanntlich in verschiedenen Tabellen verwaltet werden, aller Bücher, die zwischen 1985 und 1992 erschienen sind, anzeigen lassen. Wir wählen dazu beide Tabellen aus, die bestehende Verknüpfung wird übernommen. An dieser Stelle ist es wichtig anzumerken, daß für Abfragen neue Beziehungen zwischen Tabellen definiert werden können, die nur für die Abfrage gespeichert werden und bestehende Beziehungen zwischen Tabellen unangetastet lassen. Die Felder und Kriterien kann man der Abbildung entnehmen. Obwohl das verbindende Schlüsselelement *Buchtyp-Nr* nicht in der Abfrage auftaucht, funktioniert das Ganze einwandfrei.



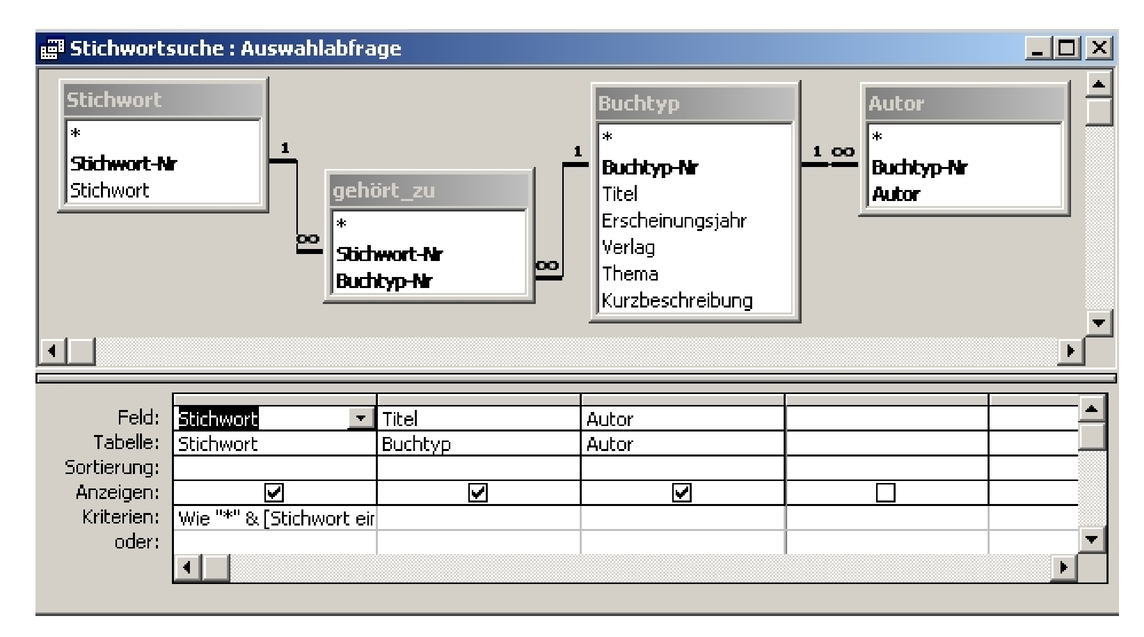
QBE-Abfrage - Autorensuche



Ein Wort noch zum Kriterium des Feldes *Autor*. Die eckigen Klammern bewirken eine so genannte Parameterabfrage, d.h. es öffnet sich ein Fenster mit dem Text "Autor eingeben", und die Abfrage wird erst nach Eingabe durchgeführt. Die WIE-Konstruktion gestattet es, z.B. nur den ersten Buchstaben einzugeben, um dann alle Bücher der mit „M“ beginnenden Autoren zu erhalten.

Unser drittes Beispiel zeigt, dass auch bei einer n:m-Verknüpfung mit Hilfe einer dritten Tabelle Abfragen ohne Probleme formuliert werden können. Dazu betrachten wir eine auch in der Praxis häufig benötigte Abfrage, die Suche nach Stichworten. Bekanntlich waren ja die Stichwörter mit einer Zuordnungstabelle über eine *Stichwort-Nr* mit einer *Buchtyp-Nr* verknüpft.

Auch hier funktioniert die Abfrage einwandfrei und liefert sogar in einer weiteren Tabelle die den Buchtypen zugeordneten Autoren.



Stichwortabfrage

Es konnten hier nur einige der Möglichkeiten angerissen werden, die *Access* für Abfragen bietet. So kann man auch neue, berechnete Feldwerte verwenden, bzw. so genannte aggregierte Berechnungen durchführen lassen. (z.B. könnte man die Anzahl der von einem Schüler ausgeliehenen Bücher durch eine solche Abfrage ermitteln.) Abfragen können mit bestimmten Aktionen verbunden werden, wie z.B. das Löschen aller Datensätze, die gewissen Kriterien gehorchen. (Eine solche Löschabfrage könnte in unserem Beispiel dazu verwendet werden, alle erledigten Ausleihvorgänge, die eine gewisse Zeit zurückliegen, aus der Tabelle Ausleihe zu eliminieren.)

Trotzdem gibt es bestimmte Auswahlen aus den Datensätzen, die sich mit einer QBE-Abfrage nicht realisieren lassen. Es handelt sich hierbei vor allem um so genannte UNION-Abfragen, bei denen Datensätze aus zwei Tabellen vereinigt werden können. Erläuterungen und Beispiele dazu finden sich in der Access-Hilfe. Hier gibt es die Möglichkeit, in Access Abfragen in SQL-Formulierung zu erstellen. Auch die gewöhnliche Abfragen, wie sie von uns in drei Beispielen vorgeführt wurden, können **in SQL formuliert** werden. Es ist auch möglich, für QBE-entworfene Abfragen die zugehörige SQL-Darstellung angeben zu lassen(Ansicht→SQL-Ansicht).

Beispielsweise

```
SELECT DISTINCTROW Buchtyp.Thema, Buchtyp.Titel, Buchtyp.Verlag
FROM Buchtyp WHERE (
((Buchtyp.Thema) Like "Programmieren" & "*") AND ((Buchtyp.Erscheinungsjahr)>1995)) OR
(((Buchtyp.Thema) Like "Programmieren" & "*") AND ((Buchtyp.Verlag)="Microsoft"))
ORDER BY Buchtyp.Titel;
```

Aufgaben



Aufgaben zu Abfragen mit QBE

Aufgabe 1

Erstellen Sie für die Informatik-Bücherei eine Abfrage, in der alle Bücher ausgegeben werden, die ein bestimmter Schüler ausgeliehen hat, mit Autor, Titel und Ausleihdatum. Eingegeben werden soll die Nummer des Ausleihers.

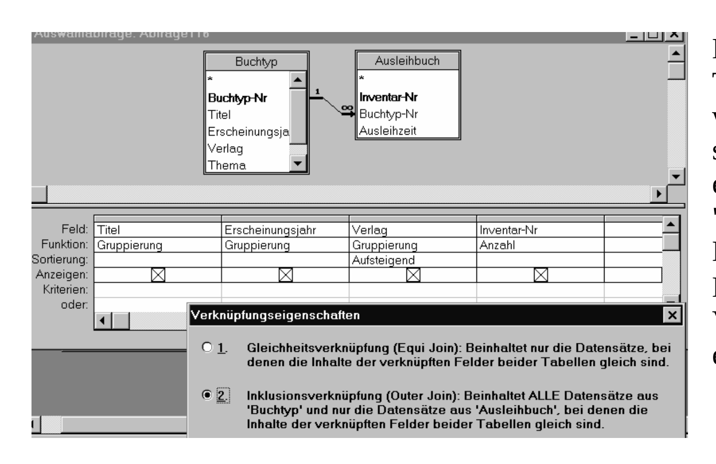
Aufgabe 2

Erstellen Sie für die Informatik –Bücherei eine Abfrage, die alle Bücher und Schüler angibt, bei denen die Ausleihfrist überschritten ist.

Aufgabe 3

Erstellen Sie für die Informatik –Bücherei eine Abfrage, die berechnet (aggregiert), wie viel Bücher eines bestimmten Buchtyps noch vorhanden sind (insgesamt und ausgeliehen).

Hilfe:
Wie viele Bücher sind von den einzelnen Buchtypen vorhanden? Mit dem Button wird in der Entwurfstabelle die Zeile „Funktion:“ eingefügt. In dieser Zeile muss bei allen Feldern, die gruppiert werden sollen das Schlüsselwort Gruppierung stehen.
Aus einem Listfeld können Sie eine der zur Verfügung stehenden Aggregatfunktionen wie Anzahl, Min, Max, Summe etc. wählen.



Damit auch die Bücher aus der Tabelle Buchtyp aufgelistet werden, die nicht ausgeliehen sind, muss die Verknüpfungs-eigenschaft von "Inner Join" auf "Outer Join" geändert werden. Machen Sie dazu einen Doppelklick auf die 1 zu ∝ Verbindung und setzen Sie die entsprechende Markierung.

Aufgabe 4

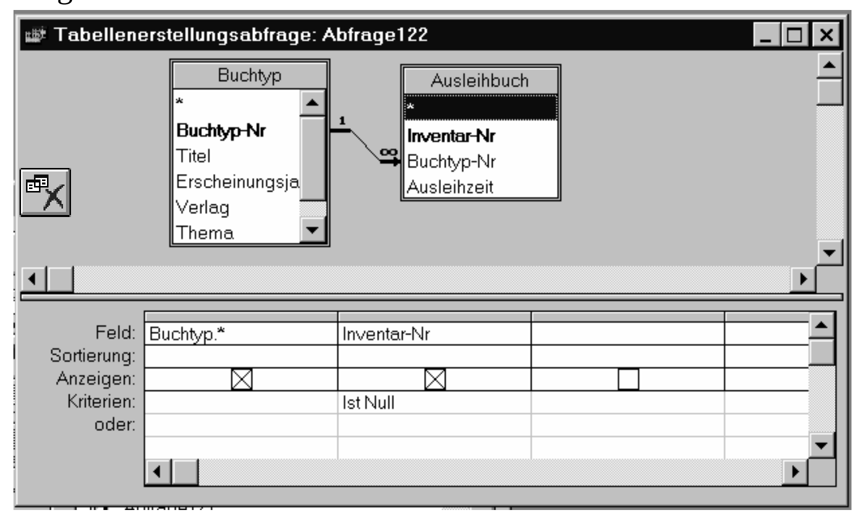
Erstellen Sie für die Informatik –Bücherei

a) eine Tabelle die alle abgeschlossenen Ausleihvorgänge umfasst, die länger als ein halbes Jahr zurückliegen.

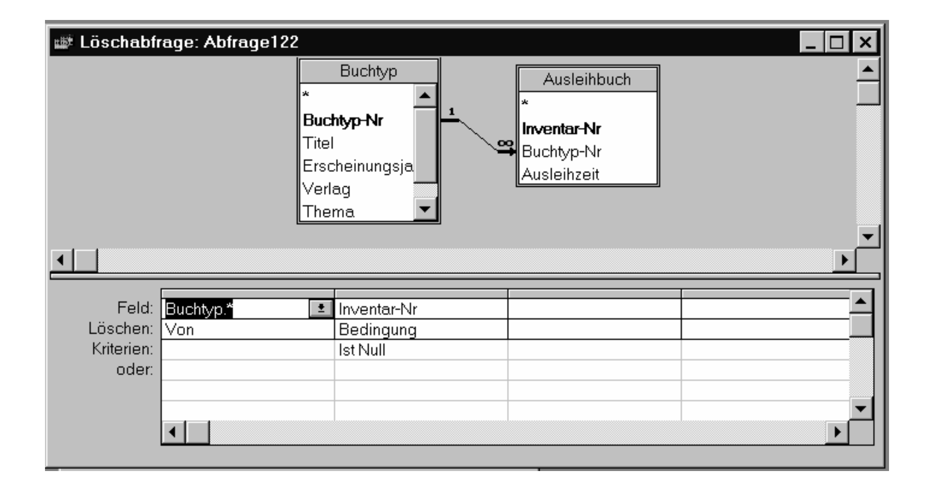
b) eine Löschabfrage, welche diese Datensätze löscht..

Hilfe:

Mit einer Tabellenerstellungsabfrage wird eine neue Tabelle aus vorhandenen Tabellen erstellt. So zum Beispiel zum Exportieren oder zum Archivieren, zum Erstellen von Sicherheitskopien oder zum Speichern von Datenbeständen.
Es sollen z.B. alle Datensätze in der Tabelle "Buchtyp", für die es keine Einträge in der Tabelle "Ausleihbuch" gibt, in einer neuen Tabelle mit dem Namen "Nichtvorhanden" kopiert werden. Dazu erstellt man zunächst eine entsprechende Auswahlabfrage mit Outer-Join und selektiert mit einem Nullwert.
Mit dem entsprechenden Button wird dann die getestete Auswahlabfrage in eine Tabellenerstellungsabfrage umgewandelt und ausgeführt.



Es sollen alle Datensätze in der Tabelle "Buchtyp" gelöscht werden, für die es keine Einträge in der Tabelle "Ausleihbuch" gibt. Wie oben beschrieben wandelt man mit dem entsprechenden Button die getestete Auswahlabfrage in eine Löschabfrage um und führt diese aus.



Aufgabe 5

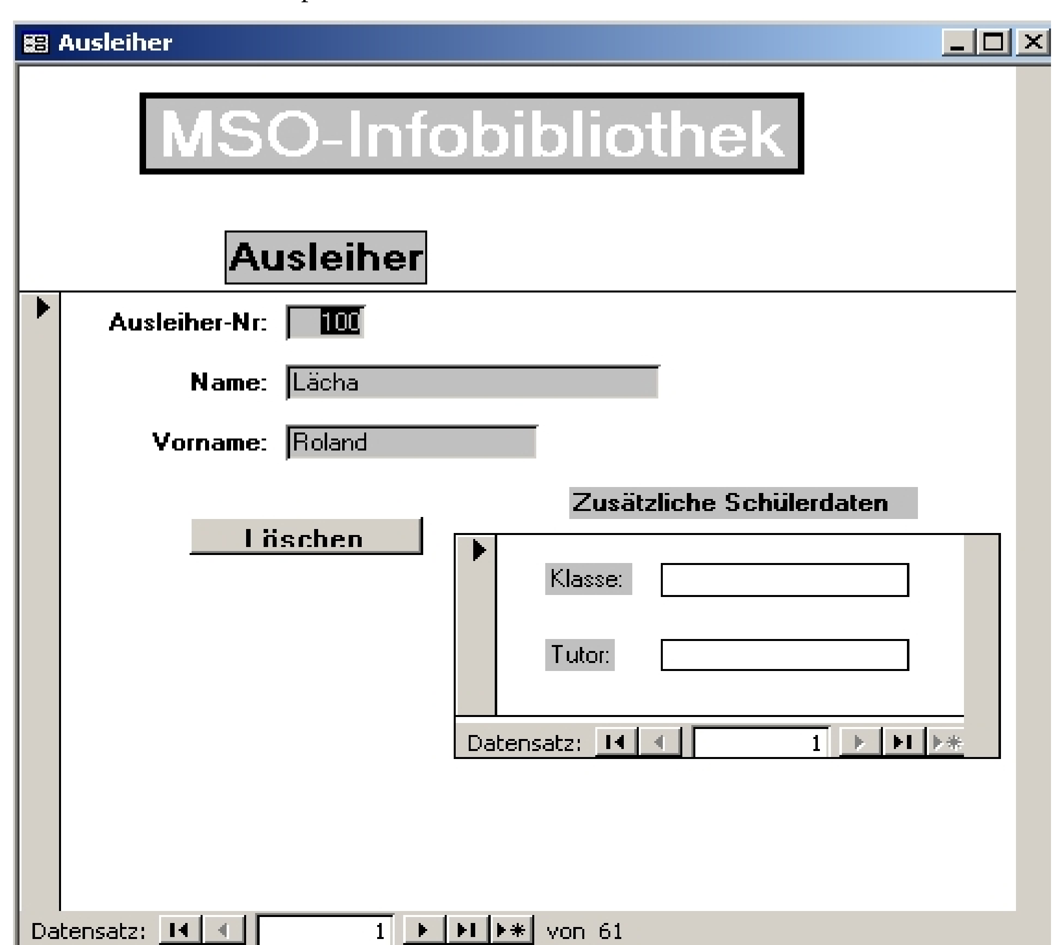
Überlegen Sie sich, welche Abfragen Sie aus Datenschutzgründen zulassen würden und für wen sie erlaubt sein sollten, bzw. denken Sie darüber nach, ob es schon bei der Konzeption der Infobücherei datenschutzrechtliche Bedenken gibt. Vergleichen Sie Ihre jetzigen Überlegungen mit denen, die Sie nach der Lektüre des Kapitels 6 über Datenschutz anstellen.

4.3 Formulare und Berichte

Im Prinzip könnten wir unsere Einführung in *Access* an dieser Stelle beenden, da sämtliche Aufgaben einer Datenbank allein mit Tabellen und Abfragen erledigt werden könnten. Mit Hilfe von Formularen ist es jedoch möglich, Eingaben wesentlich komfortabler und auch fehlersicherer zu gestalten. Stark verkürzt kann man Formulare als Eingabemasken auffassen. In der Praxis - Sie erinnern sich – realisieren Formulare die unterschiedlichen Sichten auf eine Datenbank und stellen sicher, dass nur relevante Daten gesehen werden und die Daten korrekt in den Tabellen landen.

Davon unterscheiden muss man *Berichte*, oft auch als *Reports* bezeichnet. Sie dienen im wesentlichen der Ausgabe mehrerer Datensätze, wobei es möglich ist, diese Datensätze nach bestimmten Kriterien zu gruppieren und Auswertungen über diese Datensatzgruppen durchführen zu lassen. Eine vollständige Darstellung der Möglichkeiten bei der Erstellung von Formularen und Berichten würde den Rahmen einer solchen Veröffentlichung sprengen und kann auch nicht Ziel eines Unterrichts über relationale Datenbanken sein, der mehr die Fragen der Abbildung eines realen Problems auf eine Tabellenstruktur erörtern sollte. Trotzdem soll hier an einem Beispiel das Erstellen eines Formulars angerissen werden, um eine Vorstellung von den Möglichkeiten, die *Access* in dieser Hinsicht bietet, zu bekommen.

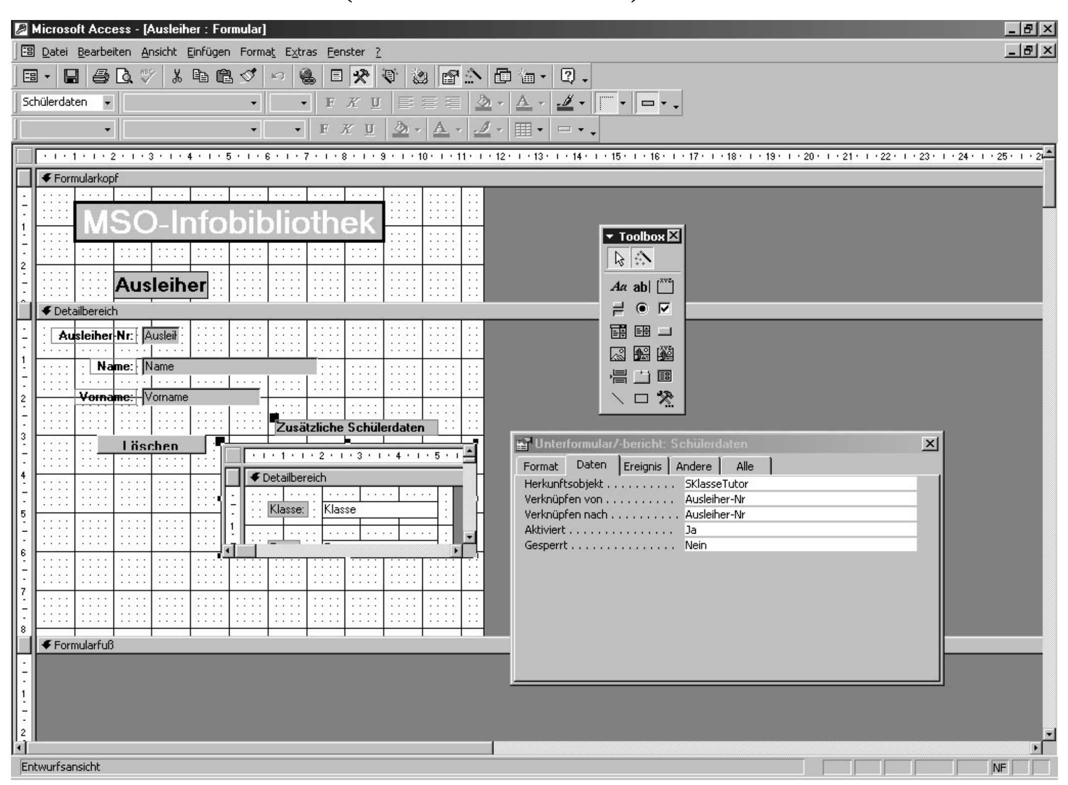
Wir wollen ein Formular für das Eingeben und Löschen von Ausleihern entwerfen. Dabei muß berücksichtigt werden, daß eine IS-A-Beziehung *Schüler-Ausleiher* noch eine zweite Tabelle notwendig machte, in der als besondere Schülerdaten *Klasse* und *Tutor* gespeichert werden. Der Bildschirmausdruck zeigt die fertige Ansicht des Formulars. Dabei soll die Taste *Löschen* ein Entfernen des angezeigten Ausleihers aus der Tabelle *Ausleiher* und eventuell aus der verknüpften Tabelle *Schüler* bewirken.



Formular - Ausleiher

Der Entwurf solcher Formulare gestaltet sich in *Access* deshalb besonders einfach, da er größtenteils mit Drag and Drop („Ziehen und Loslassen“) erfolgen kann. Zur besseren Übersichtlichkeit sind alle Bestandteile der Datenbank, wie z.B. Tabellen, Berichte und Formulare, als Objekte angelegt, die einzeln markiert werden können und deren Eigenschaften in einer Liste jederzeit zugänglich sind. Auch hier zeigt sich die Parallele zu den GUI-Buildern bei Programmiersprachen. Es soll hier natürlich keine Schritt-für-Schritt-Anleitung gegeben, sondern nur die wichtigsten Ideen vermittelt werden. Dazu ist es sinnvoll, einen Blick auf die Entwurfsansicht zu werfen, wie sie sich dem Benutzer nach Anwählen von *Formular-Entwurf* im zentralen Datenbankfenster präsentiert. Rechts oben erkennen wir eine Feldliste der zugrunde liegenden Tabelle *Ausleiher*, die beim ersten Entwurf angegeben werden muss. Die Toolbox links enthält alle Steuerelemente, die in einem Formular vorkommen können. In unserem Formular verwenden wir u.a. Textfelder mit zugehörigen Bezeichnungsfeldern. Ein Beispiel ist das Textfeld *Ausleiher-Nr*, das mit dem gleichnamigen Tabellenfeld gekoppelt ist. Links daneben findet sich das zugehörige Bezeichnungsfeld. In solchen Textfeldern erfolgen die Eingaben. Typische Eigenschaften sind Datenherkunft, Größe, Schriftart und -größe und Hintergrundfarbe. Bezeichnungsfelder tauchen auch ohne zugehörige Textfelder auf, z.B. in unserem Formular die Überschriften im Formularkopf.

Ein fortgeschrittenes Feature ist das Verwenden von Unterformularen. Dadurch ist es möglich, *Eingaben für mehrere Tabellen in einem Formular* durchzuführen. In unserem Beispiel erfolgt die Eingabe der zusätzlichen Schülerdaten *Klasse* und *Tutor* in einem solchen Unterformular. Die wichtigste Eigenschaft dieser *Unterformulare*, die *getrennt entworfen werden*, ist die Verknüpfung mit dem Hauptformular. Deshalb wurde hier der Eigenschaftseditor eingeblendet, den man, wie heute üblich, bei markiertem Objekt mit der rechten Maustaste erreicht (kontextsensitives Menü).



Eine besondere Rolle in unserem Formular spielt das Kontrollelement *Befehlsschaltfläche* mit der Aufschrift *Löschen*. Wichtigste Eigenschaft eines solchen Objekts ist die Verknüpfung mit einer Aktion. Dabei kann es sich um eine Prozedur der zugehörigen Programmiersprache *Access-Basic* oder ein so genanntes Makro handeln. In einem Makro werden Datenbankbefehle, die sonst per Hand nacheinander aufgerufen werden, automatisiert ausgeführt. Die Makros sind so wichtig, dass sie als eigenes Element neben

Tabellen, Abfragen, Formularen und Berichten im Hauptdatenbankfenster aufgeführt worden sind. Das Erstellen solcher Makros ist relativ einfach, da jeweils genaue Listen der möglichen Befehle zur Auswahl zur Verfügung gestellt werden. In unserem Fall wählen wir Befehle des Formularmenüs, in dem wir uns befinden, und zwar als erstes *Datensatz markieren* (markiert den aktuellen, im Formular angezeigten Datensatz) und zweitens *Datensatz löschen* (das löscht den markierten Datensatz). Da bei unseren Verknüpfungen die Löschweitergabe an verknüpfte Datensätze aktiviert war, wird damit auch ein zugehöriges Datum aus der Tabelle *Schüler* gelöscht. Das macht nochmals die Bedeutung von Formularen und Makros klar: Ein Benutzer muss weder über das zugrunde liegende konzeptuelle Modell und die daraus resultierenden Tabellen Bescheid wissen noch muss er mit den *Access-Basic*-Befehlen umgehen können. Ein weiteres sehr leistungsfähiges Werkzeug ist an dieser Stelle die Programmierung mittels VBA, Visual Basic for Applications, welches in allen Office-Produkten und als eigenständige Sprache (Visual Basic) verfügbar ist. Mittels VBA lassen sich ganze Anwendungen rund um die Datenbank programmieren. Bei Interesse sei auf die zahlreiche Literatur verwiesen.

Die Steuerelemente

Zum Bearbeiten von Formularen und Berichten können in der Entwurfsansicht verschiedene Steuerelemente benutzte werden.

Symbol	Name	Funktion
	Objekte markieren	Immer aktiviert. Es dient zum Markieren der und Bearbeiten der Steuerelemente.
	Steuerelement-Assistent	Wenn der Assistent aktiviert ist, wird das Einfügen von einigen Steuerelementen z. B. Listenfelder durch einen Assistenten unterstützt.
	Bezeichnungsfeld	Einfaches Feld um Beschriftungen einzufügen. Bleibt ein neues Bezeichnungsfeld ohne Eingabe, wird es beim Bestätigen oder beim Verlassen automatisch wieder entfernt.
	Textfeld	Ein Textfeld besteht aus einem Eingabe und einem Bezeichnungsfeld. Textfelder können in Formularen und Berichten Ausdrücke zum Berechnen aufnehmen. Beispiel: VK-Preis =[Nettopreis]*1,16 Berechnet den Verkaufspreis indem der Wert des Feldes „Nettopreis“ mit 1,16 multipliziert wird.
	Optionsgruppe	Um in einer Gruppe von Optionsfeldern nur die Aktivierung eines Steuerelementes zuzulassen, müssen diese in einer Optionsgruppe zusammengefasst werden. Kann auch auf Kontrollfelder angewendet werden, ist aber unüblich.
	Umschaltfläche	Diese Schaltfläche kann zwei verschiedene Zustände einnehmen und ist dadurch für die Darstellung binärer Daten geeignet z. B. Ja / Nein.
	Optionsfeld / Radiobutton	Optionsfelder werden meistens eingesetzt, um aus verschiedenen Wahlmöglichkeiten nur eine zuzulassen. Ein Optionsfeld kann zwei Zustände einnehmen.
	Kontrollkästchen	Kontrollkästen werden meistens dazu eingesetzt, um aus einer Gruppe von Möglichkeiten mehrere oder keine Optionen aktivieren zu können. Das Kontrollkästchen kann zwei Zustände einnehmen.
	Kombinationsfeld	Kombinationsfelder können zum Suchen von Datensätzen oder auch als Nachschlagefeld eingesetzt werden. Es können auch eigene Werte in ein Kombinationsfeld eingesetzt werden.
	Listenfeld	Bei einem Listenfeld können nur die enthaltenen Einträge gewählt werden.
	Befehlsschaltfläche	Schaltflächen können eingesetzt werden, um verschiedene Aktionen wie z. B. das Anwenden von Filtern oder das Aufrufen von anderen Objekten auszulösen.
	Bild	Steuerelement zum Einfügen eines Bildes.
	Objektfeld	Dient zum Einfügen von ungebundenen OLE-Objekten.
	Gebundenes Objektfeld	Zum Einfügen von gebundenen OLE-Objekten.

	Seitenwechsel	Um auf größeren Formularen den Seitenwechsel an einer bestimmten Stelle einzufügen, wird dieses Steuerelement eingefügt.
	Registersteuerelement	Durch das Registersteuerelement können die Daten eines Datensatzes in einem Formular in mehrere Register aufgeteilt werden.
	Unterformular/-bericht	Zum Einfügen eines verknüpften Unterformulares.
	Linie	Fügt eine einfache Linie ohne weitere technische Funktionen ein.
	Rechteck	Fügt ein einfaches Rechteck ohne weitere technische Funktionen ein.
	Weitere Steuerelemente	Über dieses Symbol haben Sie Zugriff auf viele weitere Steuerelemente.

Aufgaben



Aufgaben zu Formularen

Aufgabe 1

Erstellen Sie ein Formular zur Durchführung eines Ausleihvorgangs.

Aufgabe 2

Erstellen Sie ein Formular zur Eingabe eines Buchtyps, das die Eingabe von Stichworten gestattet und das automatisch die Tabelle Autor aktualisiert.

Aufgabe 3

Für besonders Fleißige und Interessierte: Ergänzen Sie die Informatik-Bücherei um all die Elemente (Abfragen, Formulare, Makros, Berichte), damit die Datenbank im Alltag funktionieren könnte.

4.4 Zusammenfassung

Die vorausgehenden Kapitel sollten eine Vorstellung davon schaffen, wie das entworfene relationale Modell mittels *Access* in eine Datenbank mit entsprechender Benutzerschnitt-stelle umgesetzt werden kann. Die Ausführungen müssen notgedrungen sich auf einzelne Beispiele beschränken und können schon gar nicht eine fundierte Einführung und Übersicht für *Access* ersetzen. Es soll vielmehr eine Anregung sein, ähnliche oder andere Probleme mit *Access* selbst umzusetzen.

Lösungsheft

Kapitel 4
–
Umsetzung des Modells mit Access

siehe Datenbankfile

Aufgaben zum Erstellen von Tabellen

Aufgabe 1

Vergleichen Sie die Datentypen in einer Programmiersprache wie Pascal mit den Datentypen von Access. Wo gibt es Gemeinsamkeiten, wo Unterschiede, wo bestehen prinzipielle Unterschiede?

Datentypen für numerische Werte und Zeichenketten existieren ganz analog in Pascal und Access, auch wenn die Wertebereiche nicht immer identisch sind. Access besitzt darüber hinaus Datentypen wie Memo oder OLE-Objekte, die datenbankspezifisch sind. Dagegen existieren in Access keine zusammengesetzten Typen oder Pointer wie in Pascal. Dateien (Tabellen) sind Access-Objekte und keine Datentypen wie in Pascal. Tabellen sind mengenorientierte Objekte, d.h. eine sequentielle Bearbeitung Datensatz für Datensatz wie in Pascal ist nicht möglich und widerspräche dem Konzept der relationalen Datenbank.

Aufgabe 2

Erstellen Sie die Tabellen für das Beispiel Schulverwaltung mit den entsprechenden Datentypen und Feldeigenschaften in Access.

do so …

Aufgabe 3

Erstellen Sie die Beziehungen für das Beispiel Schulverwaltung mit Access!

do so …

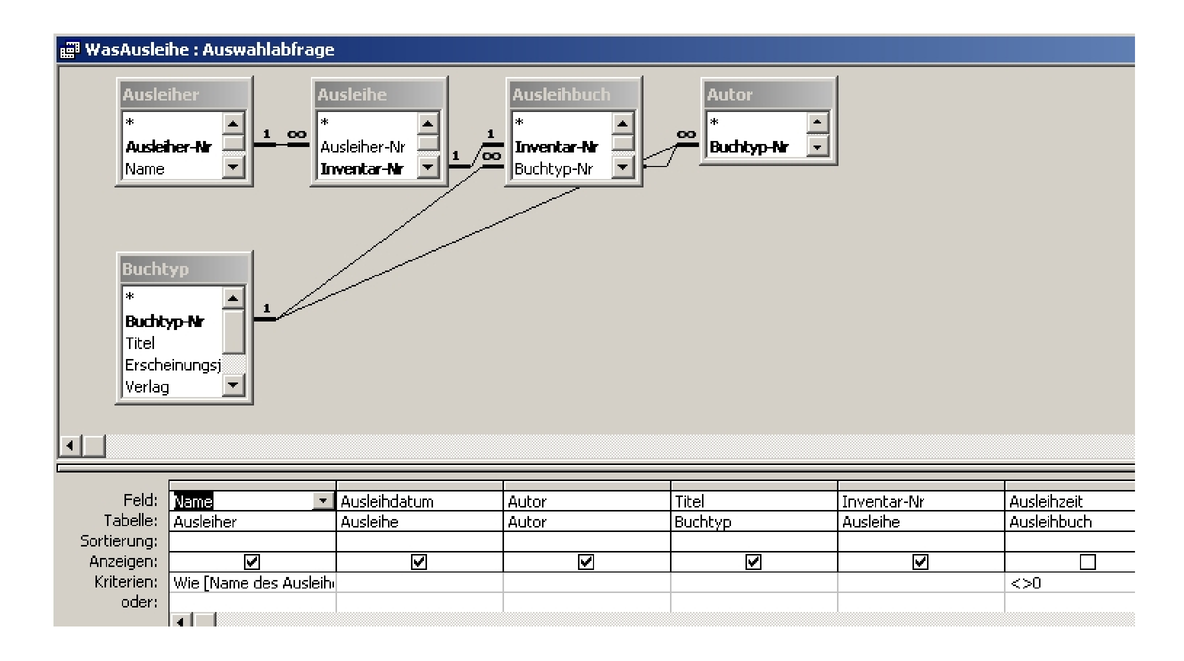
Aufgaben

Aufgaben zu Abfragen mit QBE

Aufgabe 1

Erstellen Sie für die Informatik-Bücherei eine Abfrage, in der alle Bücher ausgegeben werden, die ein bestimmter Schüler ausgeliehen hat, mit Autor, Titel und Ausleihdatum. Eingegeben werden soll die Nummer des Ausleihers.

Zuerst werden bei der Erstellung der QBE-Abfrage die beteiligten Tabellen benannt und eingefügt, die Grafik zeigt sofort die entsprechenden Verknüpfungen (Inner-Joins) an. Anschließend werden im Abfrageentwurf die beteiligten Attribute benannt, beim *Ausleiher-Namen* wird über den Ausdruckeditor die Parameterabfrage formuliert. Anschließend kann die Abfrage durchgeführt werden. Das Bild zeigt die vollständige Entwurfsansicht.



Aufgabe 2

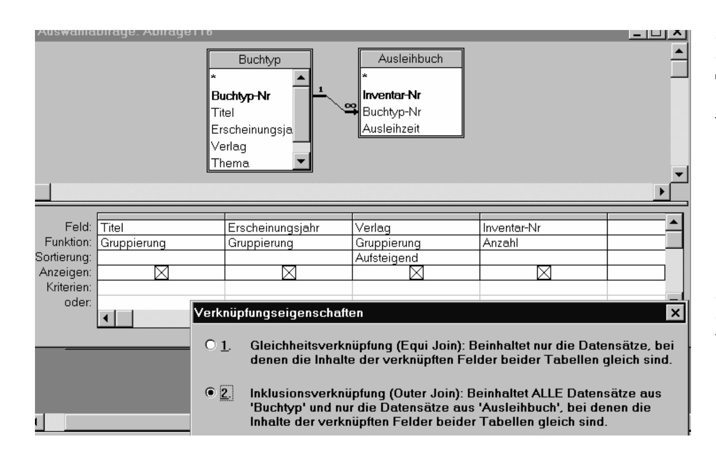
Erstellen Sie für die Informatik–Bücherei eine Abfrage, die alle Bücher und Schüler angibt, bei denen die Ausleihfrist überschritten ist.

Aufgabe 3

Erstellen Sie für die Informatik –Bücherei eine Abfrage, die berechnet (aggregiert), wie viel Bücher eines bestimmten Buchtyps noch vorhanden sind (insgesamt und ausgeliehen).

Hilfe:
Wie viele Bücher sind von den einzelnen Buchtypen vorhanden? Mit dem Button Σ wird in der Entwurfstabelle die Zeile „Funktion:“ eingefügt. In dieser Zeile muss

bei allen Feldern, die gruppiert werden sollen das Schlüsselwort Gruppierung stehen.
Aus einem Listfeld können Sie eine der zur Verfügung stehenden Aggregatfunktionen wie Anzahl, Min, Max, Summe etc. wählen.



Damit auch die Bücher aus der Tabelle Buchtyp aufgelistet werden, die nicht ausgeliehen sind, muss die Verknüpfungseigenschaft von "Inner Join" auf "Outer Join" geändert werden. Machen Sie dazu einen Doppelklick auf die 1 zu $\propto$ Verbindung und setzen Sie die entsprechende Markierung.

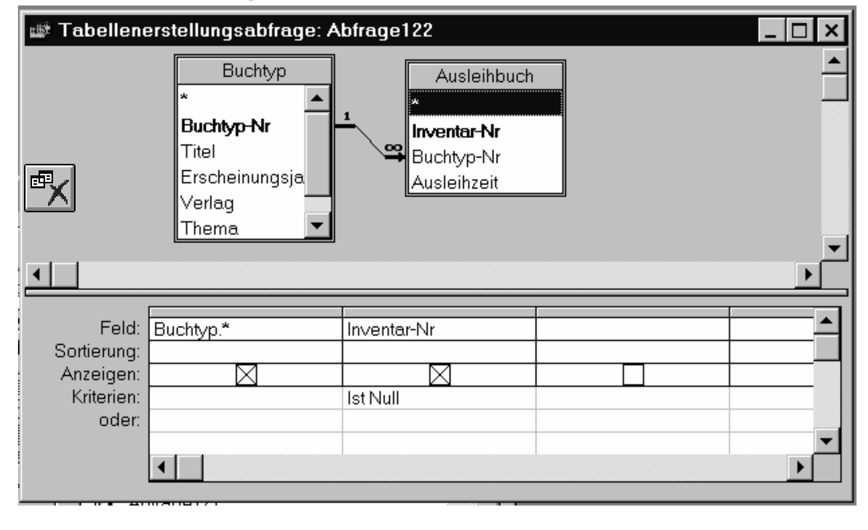
Aufgabe 4

Erstellen Sie für die Informatik –Bücherei

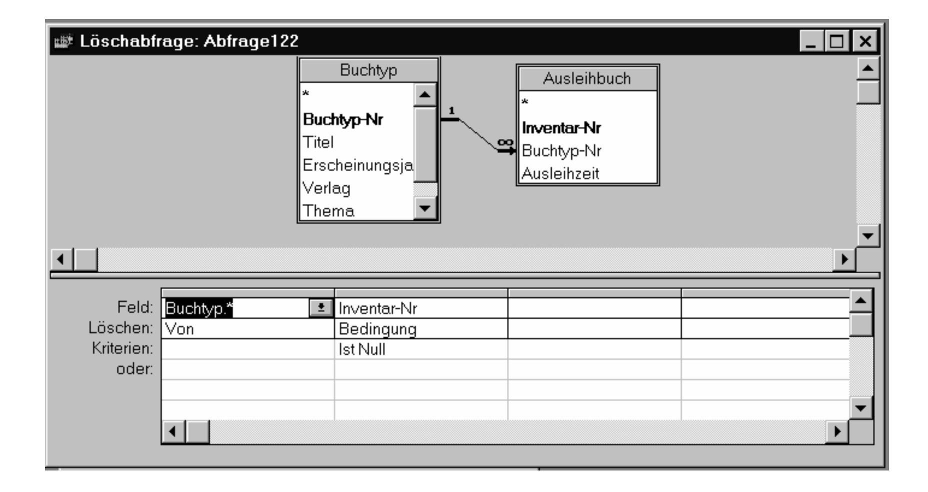
a) eine Tabelle die alle abgeschlossenen Ausleihvorgänge umfasst, die länger als ein halbes Jahr zurückliegen.

b) eine Löschabfrage, welche diese Datensätze löscht..

Hilfe:
Mit einer Tabellenerstellungsabfrage wird eine neue Tabelle aus vorhandenen Tabellen erstellt. So zum Beispiel zum Exportieren oder zum Archivieren, zum Erstellen von Sicherheitskopien oder zum Speichern von Datenbeständen.
Es sollen z.B. alle Datensätze in der Tabelle "Buchtyp", für die es keine Einträge in der Tabelle "Ausleihbuch" gibt, in einer neuen Tabelle mit dem Namen "Nichtvorhanden" kopiert werden. Dazu erstellt man zunächst eine entsprechende Auswahlabfrage mit Outer-Join und selektiert mit einem Nullwert.
Mit dem entsprechenden Button wird dann die getestete Auswahlabfrage in eine Tabellenerstellungsabfrage umgewandelt und ausgeführt.



Es sollen alle Datensätze in der Tabelle "Buchtyp" gelöscht werden, für die es keine Einträge in der Tabelle "Ausleihbuch" gibt. Wie oben beschrieben wandelt man mit dem entsprechenden Button die getestete Auswahlabfrage in eine Löschabfrage um und führt diese aus.



Aufgabe 5

Überlegen Sie sich, welche Abfragen Sie aus Datenschutzgründen zulassen würden und für wen sie erlaubt sein sollten, bzw. denken Sie darüber nach, ob es schon bei der Konzeption der Infobücherei datenschutzrechtliche Bedenken gibt. Vergleichen Sie Ihre jetzigen Überlegungen mit denen, die Sie nach der Lektüre des Kapitels 6 über Datenschutz anstellen.

Aufgaben

Aufgaben zu Formularen

Aufgabe 1

Erstellen Sie ein Formular zur Durchführung eines Ausleihvorgangs.

Aufgabe 2

Erstellen Sie ein Formular zur Eingabe eines Buchtyps, das die Eingabe von Stichworten gestattet und das automatisch die Tabelle Autor aktualisiert.

Aufgabe 3

Für besonders Fleißige und Interessierte: Ergänzen Sie die Informatik-Bücherei um all die Elemente (Abfragen, Formulare, Makros, Berichte), damit die Datenbank im Alltag funktionieren könnte.

5 Abfragen mit SQL

Entscheidend für die Qualität einer Datenbank ist, wie die in den Tabellen „steckenden“ Informationen wieder zurück gewonnen und mit anderen Informationen verknüpft werden können. Das relationale Modell stellt hierfür den theoretischen Hintergrund bereit. Die Umsetzung der relationalen Operationen in eine *Abfragesprache* ist allerdings je nach Einsatz spezieller Datenbanksoftware unterschiedlich realisiert.

Insofern ist es vor allem für den Ausbildungsbereich sinnvoll, sich eines Quasi-Standards zu bedienen, der in den letzten Jahren noch an Bedeutung hinzugewonnen hat: *SQL (Structured Query Language)*. SQL ist ursprünglich im Rahmen eines IBM-Forschungsprojektes entwickelt worden, wurde aber 1986 als Standard der amerikanischen Normbehörde (ANSI) übernommen und 1987 auch international genormt (ISO). SQL lässt sich innerhalb der Programmiersprachen als Sprache der 4. Generation bezeichnen (*4GL)*, denn sie ist im Gegensatz zu den prozeduralen Sprachen der 3. Generation, die durch Datenstrukturen und Steuerstrukturen gekennzeichnet und bei Dateien satzorientiert sind, nicht-prozedural und mengenorientiert. Ein kleines Beispiel zur Ausgabe aller Titel und Themen aus der Datei BUCHTYP, die vor 1990 erschienen sind, möge dies verdeutlichen:

problemorientierte Sprache (3GL)	SQL (4GL)

open (buchtyp)
while not eof(buchtyp)
{
  read(buch);
  if (buch.jahr < 1990)
  print(buch.autor,buch.titel);
}
	select titel,thema
	from buchtyp
	where erscheinungsjahr < 1990

SQL ist allerdings keine universelle Programmiersprache wie z.B. C oder Pascal, sondern beinhaltet im wesentlichen nur Befehle zur Verwaltung der Datenbank. Viele SQL-Implementationen erlauben deshalb auch, SQL-Statements in eine Programmiersprache einzubetten, so z.B. *Java*. Umgekehrt ist es heute möglich, innerhalb der höheren Programmiersprachen wie z.B. Java oder Pascal/Delphi SQL-Statements zu formulieren und an eine Datenbank zu richten.

Die Gesamtheit der SQL-Befehle wird gewöhnlich in drei Klassen eingeteilt:

- DDL-Befehle zur Definition von Datenstrukturen (Tabellen, Indizes)
- DCL-Befehle zur Kontrolle der Zugriffsberechtigungen
- DML-Befehle zur Manipulation der Datenbank und Formulierung von Abfragen.

Wir wollen uns nur mit den DML-Befehlen beschäftigen, weil sie den wesentlichen Kern von SQL ausmachen und *Access* alle Abfragen hierüber realisiert. Die DDL-Befehle sind äußerst unhandlich, weswegen die meisten SQL-Datenbanken komfortablere Methoden zur Verfügung stellen. In *Access* lassen sich die DDL-Befehle wie CREATE, TABLE, ALTER, DROP, INDEX mit der zugehörigen Programmiersprache VBA einsetzen oder aber im Menü *Abfrage* unter dem Stichwort *SQL-spezifisch* zur Datendefinition verwenden, was hier aber nicht betrachtet werden soll.

5.1 Auswahl-Abfragen mit SELECT

Eine **Auswahl-Abfrage** mit SQL dient dazu, die relationalen Operationen ***Projektion*** und ***Selektion* zu realisieren**; wenn die Abfrage sich auf *mehrere Tabellen* bezieht, wird vorher ein *Join* durchgeführt. Das Ergebnis ist nach dem relationalen Modell wieder eine Tabelle, die natürlich aus nur einer Zeile und Spalte bestehen oder sogar leer sein kann. Diese Tabelle wird bei einer Auswahl-Abfrage nicht etwa neu angelegt, sondern stellt nur eine Sicht auf die Datenbank dar.

Die Syntax des Standard-SELECT-Befehls (im Backus-Naur-Formalismus unten dargestellt), mit dem solche Abfragen in SQL formuliert werden können, ist recht kompliziert, vor allem wegen der theoretisch beliebigen Schachtelungstiefe durch Unterabfragen:

```
SELECT [ALL|DISTINCT]{spalten|*}
FROM tabelle [alias] [tabelle[alias]]...
[WHERE {bedingung|unterabfrage}]
[GROUP BY spalten[HAVING {bedingung|unterabfrage}]]
[ORDER BY spalten[ASC|DESC]...];
```

Aus dieser Darstellung ergibt sich, daß eine einfache SQL-Abfrage an unsere Datenbank wie

```
SELECT *
FROM buchtyp;
```

uns alle Zeilen und Spalten der Tabelle *Buchtyp* liefert, dabei steht das Symbol * für “alle Spalten der Tabelle“. Werden anstatt des Symbols * Spaltennamen (Attribute) angegeben, liefert der Befehl nur diese Spalten (**Projektion**). Die **WHERE-Klausel** erzeugt eine **Selektion**, wie wir bereits im Beispiel auf der vorherigen Seite gesehen haben.

Wie ein SQL-Interpreter einen speziellen Befehl abarbeitet, ist für den Anwender uninteressant. Zum besseren Verständnis ist es aber sinnvoll, die einzelnen Klauseln in einer bestimmten Reihenfolge zu betrachten, um sich selbst die Arbeitsweise zu veranschaulichen. Hierbei ist folgendes Modell hilfreich:

Klausel	Betrachtungsschritt	Erläuterungen
select	6	streicht alle nicht genannten Spalten
from	1	bildet das Kreuzprodukt der genannten Tabellen
where	2	streicht alle Zeilen, welche die Bedingung nicht erfüllen
group by	3	bildet Gruppen
having	4	streicht alle Zeilen, welche die Bedingung nicht erfüllen (nur mit Gruppierung verwendbar)
order by	5	sortiert auf- oder absteigend

Nach diesem Modell ist es also sinnvoll, sich zuerst die beteiligten Tabellen anzusehen, bei mehreren beteiligten Tabellen den Join durchzuführen und anschließend die Selektionen und Anordnungen zu betrachten. Ganz zuletzt ist dann die Projektion auf die benannten Spalten durchzuführen.

5.1.1 Einfache Auswahl-Abfragen

In *Access* werden - wie wir im letzten Kapitel gesehen haben - Abfragen meist im QBE-Modus erstellt und durchgeführt. *Access* setzt diese aber intern in SQL um, damit auch SQL-Datenbanken eingebunden werden können. In der Symbolleiste *Ansichten* besteht die Möglichkeit, die mit der QBE-Technik erstellte Abfrage als SQL-Befehl anzusehen. Wir werden noch sehen, dass Abfragen auch direkt als SQL-Statement eingegeben werden können, ohne die QBE-Technik zu verwenden. Dies hat sogar den Vorteil, daß man sich standardisierter Techniken bedient und sich die Fehlermeldungen von *Access* immer auf die SQL-Formulierung beziehen, also direkt ausgewertet werden können.

In Kapitel 4 haben wir eine Auswahl-Abfrage auf die Tabelle *Buchtyp* ausgeführt, um Bücher zum Thema "Programmieren" zu finden. Wählt man bei geöffneter Abfrage im Untermenü *Ansicht* den Menüpunkt SQL, erhält man das folgende SQL-Statement, das von *Access* automatisch generiert wurde, in einem speziellen Fenster:

Themensuche : Auswahlabfrage

```
SELECT DISTINCTROW Buchtyp.Thema, Buchtyp.Titel, Buchtyp.Verlag
FROM Buchtyp
WHERE ((Buchtyp.Thema Like "Programmieren" & "*") AND (Buchtyp.Erscheinungsjahr>1995)) OR ((Buchtyp.Thema Like
"Programmieren" & "*") AND (Buchtyp.Verlag="Microsoft"))
ORDER BY Buchtyp.Titel;
```

SQL-Darstellung einer (Auswahl)Abfrage

Mittels unserer o.a. Schrittmethode zum Verständnis des SQL-Befehls lässt dieser sich ganz leicht verstehen:

1. Die Tabelle *Buchtyp* wird bearbeitet (FROM)
2. Es werden diejenigen Zeilen (Datensätze) gesucht, welche die Bedingung hinter WHERE erfüllen. Diese Bedingung ist aus verschiedenen Unterbedingungen zusammengesetzt, die mit AND oder OR verknüpft sind (SELECT...WHERE).
3. Die gefundenen Datensätze werden nach dem Titel alphabetisch geordnet
4. Ausgegeben werden die Spalten (Attribute) Thema, Titel, Verlag. Das DISTINCTROW verhindert die Ausgabe doppelter Datensätze (in *Access* anstatt des Standards DISTINCT).

Die Schreibweise *Buchtyp.Thema* lässt erkennen, dass das Attribut *Thema* der Tabelle *Buchtyp* gemeint ist. Diese genaue Spezifizierung ist syntaktisch nicht notwendig, erleichtert aber die Übersicht.

<u>Anmerkung</u>: Auch hier erkennen Sie wieder die Parallele zu den objektorientierten Programmiersprachen. Es handelt sich nämlich um die Darstelllung *Objekt.Methode*.

Wir wollen nun eine Abfrage direkt in SQL formulieren, ohne den Umweg über QBE zu gehen. Uns interessiert, wie viele Bücher die verschiedenen Verlage in unserer Bibliothek stellen. Hierzu klicken wir den Button *Neu* bei den Objekten *Abfragen* an und wählen in der Ansicht den Menüpunkt SQL. Der SQL-Editor stellt das bereits bekannte Fenster zur

Verfügung. Haben wir vorher bereits die Tabelle *Buchtyp* ausgewählt, so ist das Grundgerüst unserer Abfrage bereits im Editor-Fenster zu sehen:

Anzahl	Verlag
24	
6	Addison-Wesley
4	Autodesk
2	BI
1	Birkhäuser
41	Borland
2	Central Point
2	Corel
1	Digitalk
2	Dümmler
2	Hanser
1	Heise
1	IBM
1	IWT
2	Klett
1	Luther
2	Markt+Technik
2	McGrawHill
22	Microsoft
4	Microsoft-Press
14	Novell
1	Schöningh
2	Spektrum-
3	Springer
3	Sybex
4	Teubner
2	TeWi
1	VDI
1	Vieweg
1	Vogel
2	Würfel

```
SELECT DISTINCTROW
FROM Buchtyp;
```

Die Anzahl-Funktion COUNT lässt sich zur Gruppierung verwenden, so dass wir ergänzen können:

```
SELECT DISTINCTROW
    COUNT(Titel)
    AS Anzahl, Verlag
  FROM Buchtyp
  GROUP BY Verlag;
```

Nach dem Aufruf des Menüpunktes *Ausführen* im Menüpunkt *Abfragen* wird in der *Datenblattansicht* das Ergebnis der Abfrage angezeigt. Sie lässt erkennen, dass wir mit `AS Anzahl` eine Spaltenüberschrift definiert haben, die zweite Spalte zeigt die Verlagsnamen an. Die NULL-Einträge bei 24 Büchern lassen sich durch einen Blick in die Tabelle *Buchtyp* leicht erklären: Das sind Bücher, für die wir bisher in der Ausgangstabelle noch keinen Verlagsnamen eingetragen haben - unsere Datenbank besitzt demnach noch einige „Löcher“, die noch zu schließen sind.

Wollen wir die Ausgabe der Buchanzahlen ohne Verlagsangabe unterdrücken, führt die Hinzunahme der HAVING-Klausel zum Ziel:

```
HAVING Verlag IS NOT NULL
```

<u>Weitere Beispiele</u>:

Die folgende SQL-Anweisung liefert anhand der Tabelle *Kurs* (vgl. S. 42) die Kurse der Jahrgangsstufe 12 geordnet nach Lehrernummer:

```
SELECT DISTINCTROW KursNr, Thema, LehrerNr
FROM Kurs
WHERE (Jahrgangsstufe=“12/I“) OR
     (Jahrgangsstufe =“12/II“)
ORDER BY LehrerNr;
```

Die nicht ausgeliehenen Spinde aus der Tabelle *Spind* erhält man so:

```
SELECT DISTINCTROW SpindNr, Standort
FROM Spind
WHERE SchülerNr IS NULL;
```

5.1.2 Auswahl-Abfragen mit Join

Bisher war an der Auswahl-Abfrage nur eine Tabelle beteiligt. Im letzten Kapitel haben wir aber bereits eine QBE-Abfrage formuliert, die uns *Autor* und *Titel* aller Bücher liefert, die zwischen 1985 und 1992 erschienen sind. Die SQL-Abfrage lässt sich wieder anzeigen und offenbart, dass die beiden Tabellen *Autor* und *Buchtyp* durch einen Join verbunden sind, da die Attribute *Autor* und *Titel* in verschiedenen Tabellen verwaltet werden:

```
SELECT DISTINCTROW Autor.Autor, Buchtyp.Titel,
  Buchtyp.Erscheinungsjahr
FROM Buchtyp INNER JOIN Autor
ON Buchtyp.[Buchtyp-Nr]=Autor.[Buchtyp-Nr]
WHERE (((Autor.Autor) LIKE [Autor eingeben]&“*“) AND
  ((Buchtyp.Erscheinungsjahr)>1985 AND
  (Buchtyp.Erscheinungsjahr)>1992))
ORDER BY
  Autor.Autor,Buchtyp.Titel,Buchtyp.Erscheinungsjahr;
```

Wir haben bei der Erstellung der Datenbank bereits die Beziehung zwischen den Tabellen *Autor* und *Buchtyp* über das gemeinsame Attribut *Buchtyp-Nr* definiert. *Access* macht hieraus einen **Equi-Join**, der alle Datensätze der beiden beteiligten Tabellen mit gleichen Werten für das gemeinsame Attribut umfasst. Die obige Schreibweise für diesen Join ist nicht SQL-Standard, nach dem nur die beiden Tabellen angegeben werden müssen. Dies rührt vor allem daher, dass *Access* auch noch einen *Outer-Join* zur Verfügung stellt, der aber hier nicht weiter behandelt werden soll (vgl. Übungsaufgabe). Die übrigen Teile der SQL-Abfrage erklären sich anhand der QBE-Technik, die wir im vorigen Kapitel behandelt haben. Eine Besonderheit am Rande: *Access* setzt zusammengesetzte Bezeichner wie *Buchtyp-Nr* in eckige Klammern, was zu vielen syntaktischen Fehlern Anlaß geben kann - solche Bezeichner sollten in der SQL-Variante vermieden bzw. durch *BuchtypNr* ersetzt werden. So sind auch Verwechslungen mit der anderen Verwendung der eckigen Klammern zum Kennzeichnen von Variablenwerten, wie z.B. bei der Parameterabfrage `[Autor eingeben]` ausgeschlossen.

Wir wollen nun eine eigene kleine SQL-Abfrage erstellen, die zwei Tabellen umfaßt. Wir wollen die Namen aller Ausleiher wissen:

```
SELECT Name, Vorname
FROM Ausleiher INNER JOIN Ausleihe ON
   Ausleiher.[Ausleiher-Nr] = Ausleihe.[Ausleiher-Nr]
ORDER BY Name, Vorname;
```

Auch hier wird der Join verwendet; es werden also alle Datensätze der Gleichheitsbedingung nach dem Schlüsselwort `ON` gesucht. Da ein Ausleiher in der Bücherei auch mehrere Bücher ausleihen kann, fehlt im SQL-Befehl das DISTINCTROW: Die entsprechenden Ausleiher treten im Ergebnis auch mehrfach auf.

Wie wir bereits im letzten Kapitel gesehen haben, lassen sich Abfragen auch auf mehr als zwei Tabellen ausdehnen. Wir hatten die Suche nach Büchern behandelt, für die in der Tabelle *Stichwort* ein bestimmtes Stichwort gespeichert ist. Da es sich um eine n:m-Beziehung handelt, sind zusätzlich die Tabellen g*ehört_zu* und *Buchtyp* angesprochen; den Autor erhalten wir durch die Beziehung mit der Tabelle *Autor*. Die entsprechende QBE-Abfrage (siehe Kapitel 4) setzt *Access* folgendermaßen um:

```
SELECT DISTINCTROW Stichwort.Stichwort,
   Buchtyp.Titel, Autor.Autor
FROM Stichwort INNER JOIN ((Buchtyp INNER JOIN Autor
   ON Buchtyp.[Buchtyp-Nr] = Autor.[Buchtyp-Nr])
   INNER JOIN gehört_zu ON Buchtyp.[Buchtyp-Nr] =
   gehört_zu.[Buchtyp-Nr]) ON Stichwort.[Stichwort-
   Nr] = gehört_zu.[Stichwort-Nr]
WHERE (((Stichwort.Stichwort) Like "*" & [Stichwort
   eingeben] & "*"));
```

Dieses Statement ist mit den bekannten Regeln sofort verständlich. Die Reihenfolge der Joins ergibt sich aus den bereits definierten Beziehungen zwischen den beteiligten Tabellen und läßt sich an der Klammerschachtelung erkennen: Zuerst wird *Buchtyp* mit *Autor* gejoint, das Ergebnis mit *gehört_zu* und dieses Ergebnis mit der Tabelle *Stichwort.*

5.1.3 Unterabfragen

SQL-Anweisungen lassen sich auch schachteln. Eine Unterabfrage kann in der Attributliste der SELECT-Anweisung und als Vergleichsoperand in der WHERE- oder HAVING-Klausel verwendet werden. Dabei muss – außer bei speziellen Ausnahmen, auf die hier nicht eingegangen wird – das Ergebnis der Unterabfrage genau einen Wert zurückliefern. die Unterabfrage wird syntaktisch in runde Klammern gesetzt und vom Interpreter zuerst ausgewertet. Da in der Unterabfrage in der FROM- Klausel auch eine andere Tabelle als in der Hauptabfrage benannt werden kann, ist damit eine – allerdings unübersichtlichere – Alternative zum Join gegeben.

Würde unsere Datenbank in der Tabelle *Buchtyp* noch das Attribut *Ausleihzahl* enthalten, um eine Ausleihstatistik aufzubauen, könnte der folgende SQL- Befehl dazu dienen, die überdurchschnittlich ausgeliehenen Bücher herauszufinden (die Funktion AVG liefert den Mittelwert):

```
SELECT Titel, Ausleihzahl
FROM Buchtyp
WHERE Ausleihzahl>(SELECT AVG (Ausleihzahl)
FROM Buchtyp);
```

Um die ausgeliehenen Bücher mit den Daten zu finden, müssen die Tabellen *Ausleihbuch* und *Ausleihe* miteinander verknüpft werden. Die SQL-Abfrage mit Unterabfrage zur Ausgabe der ausgeliehenen Bücher verwendet die IN-Klausel, damit das Ergebnis der Unterabfrage eindeutig ist:

```
SELECT DISTINCTROW *
FROM Ausleihbuch
WHERE [ Inventar-Nr] IN
     (SELECT [ Inventar-Nr]
     FROM Ausleihe
     WHERE Ausleihbuch.[Inventar- Nr]
     Ausleihe. [ Inventar-Nr]));
```

Ein analoges Ergebnis liefert der Join der beiden Tabellen mit dem zusätzlichen Vorteil, dass auf die Attribute beider Tabellen zurückgegriffen werden kann:

```
SELECT DISTINCTROW *
FROM Ausleihbuch INNER JOIN Ausleihe ON
     Ausleihbuch.[Inventar- Nr]=
     Ausleihe.[ Inventar-Nr];
```

5.2 Aktionsabfragen

Bislang dienten die SQL-Befehle dazu, Abfragen zu formulieren und damit Anfragen an die Datenbank zu richten. Die Ergebnistabellen stellen eine virtuelle Sicht des Benutzers auf die Datenbank dar, weil die Ergebnistabellen nicht als externe Dateien gespeichert werden. Dies ist nach dem relationalen Datenbankmodell auch nicht sinnvoll, da sonst redundante Daten erzeugt würden. In der betrieblichen Realität sind allerdings bestimmte Abfragen das tägliche Brot der Datenbank-Anwendung und hier kann es ohne weiteres sinnvoll sein, nicht nur die Abfrage zu speichern, sondern aus Zeitgründen auch die fertige Ergebnistabelle. Mit SQL lässt sich die Ergebnistabelle direkt anlegen, allerdings in Access – anders als in Standard-SQL mit CREATE TABLE ...AS – mit SELECT ... INTO:

```
SELECT *
INTO Neubuch
FROM Buchtyp
WHERE Erscheinungsjahr > 2000;
```

Wesentlich wichtiger ist die Möglichkeit, aufgrund einer Abfrage die Datenbank zu aktualisieren. Denkbar wäre in einer Artikeldatenbank die Umstellung des Preises von DM in EURO nach dem festgelegten Umrechnungsschlüssel oder die Veränderung des Mehrwertsteuersatzes nach einer Steuererhöhung. In unserer Bücherei könnte die Aktualisierung dazu verwendet werden, Ausleiher-Nummern aus datenschutzrechtlichen Gründen zu löschen. Hierzu ist ein Update der Tabelle *Ausleihe* bei der Rückgabe des Buches oder am Schuljahresende sinnvoll:

```
PDATE Ausleihe
ET [Ausleiher-Nr]  = 999
HERE [Ausleiher-Nr] LIKE [Inventarnummer: ];
```

Selbstverständlich ist auch das Löschen des gesamten Datensatzes in der Tabelle *Ausleihe* möglich, wenn das Buch zurückgegeben wird:

```
ELETE FROM Ausleihe
HERE [Ausleiher-Nr] LIKE [Inventarnummer eingeben: ];
```

Um eine Ausleihstatistik zu erhalten, könnte ein zusätzliches Attribut *Ausleihzahl* in der Tabelle *Buchtyp* um 1 erhöht werden, wenn eine Ausleihe erfolgt; auch dies ist mit UPDATE realisierbar. In der Praxis setzt man zum Update der Datenbank bei Bücherausleihe oder Rückgabe Formulare ein, weil dieser benutzerfreundlicher zu verwenden sind als die Formulierung als SQL-Abfragen.

Aufgaben

Aufgaben zu SQL-Abfragen

Aufgabe 1

Gegeben ist eine Datenbank mit folgenden Relationen:

```
LIEFERER(LiefererNr,LieferName,Strasse,Plz,Ort,Rabatt)
ARTIKEL(ArtikelNr,Artikel,Preis,GruppenNr,Meldebestand,
          Bestand)
ARTIKELLIEFERER(ArtikelNr,LiefererNr,BestellNr,
          Angebotspreis)
WARENGRUPPE(GruppenNr,Gruppe)
```

a) Was bewirkt folgende Anweisung?

```
SELECT DISTINCTROW ArtikelLieferer.LieferNr,
     ArtikelLieferer.ArtikelNr,Lieferer.LieferName
FROM Lieferer INNER JOIN ArtikelLieferer
ON Lieferer.LieferNr=ArtikelLieferer.LieferNr
WHERE ArtikelLieferer.ArtikelNr=1616;
```

b) Eine Lieferliste ist zu erstellen, die nach Lieferer-Ort geordnet ist!

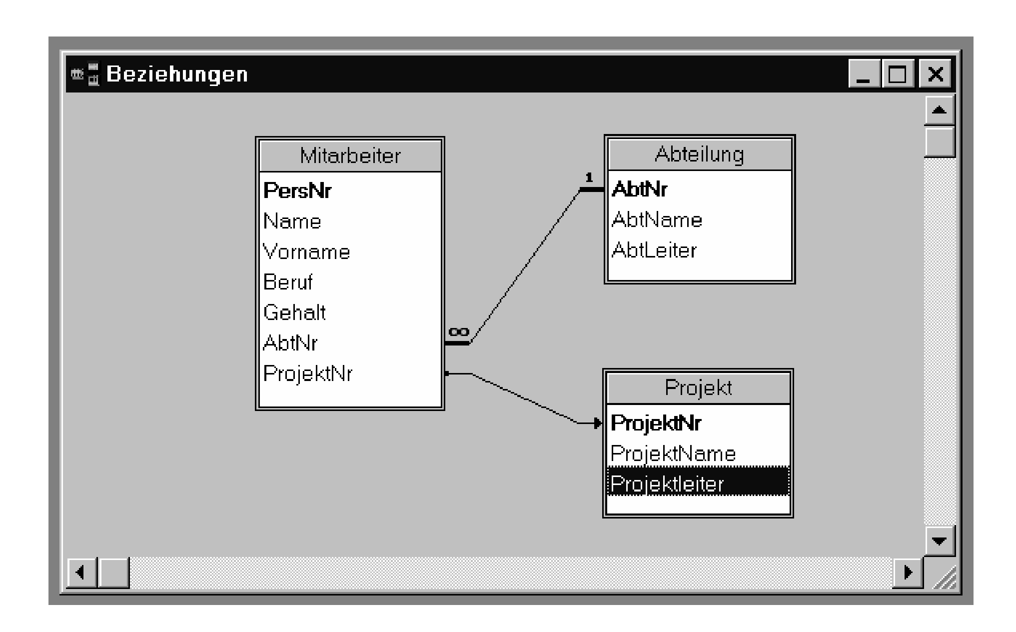
c) Was bewirkt die folgende Aktionsabfrage?

```
UPDATE DISTINCTROW ArtikelLieferer
SET ArtikelLieferer.Angebotspreis=Angebotspreis*0,97
WHERE ArtikelLieferer.LiefererNr = 12345;
```

d) Welche Lieferanten gewähren einen Rabatt, der niedriger ist als der durchschnittliche Rabattsatz?

Aufgabe 2

Gegeben ist die folgende Datenbank:



In einer Firma gibt es Mitarbeiter. Jeder Mitarbeiter ist eindeutig einer Abteilung zugeordnet. Manche Mitarbeiter sind an einem Projekt beteiligt. Ein Mitarbeiter darf nur an einem Projekt arbeiten. Jede Abteilung hat einen Abteilungsleiter und jedes Projekt hat einen Projektleiter. Projektleiter und Abteilungsleiter sind natürlich auch Mitarbeiter.

Testen Sie die folgenden SQL-Befehle nach Erstellen der Datenbank und beschreiben Sie ihre Wirkung:

a) zu Projektion und Selektion

```
SELECT *
    FROM Mitarbeiter
    WHERE PersNr = 28

SELECT PersNr, Name, Vorname
    FROM Mitarbeiter
    WHERE AbtNr = 20
    ORDER BY Name, Vorname
```

b) zu logischen Verknüpfungen und Wildcards

```
SELECT PersNr, Name, Vorname
    FROM Mitarbeiter
    WHERE Name IN ("Fritsch", "Schuster","Roller")

SELECT PersNr, Name, Vorname, Gehalt
```

```
     FROM Mitarbeiter
     WHERE Gehalt BETWEEN 2000 AND 3000

SELECT PersNr, Name, Vorname
     FROM Mitarbeiter
     WHERE Name LIKE  "Me*"

SELECT PersNr, Name, Vorname
     FROM Mitarbeiter
     WHERE ProjektNr is Null
```

c) zur Ausgabe und zur Gruppierung

```
SELECT Mitarbeiter.PersNr, Mitarbeiter.Name AS [Name des
Mitarbeiters], Mitarbeiter.Gehalt AS Monatsgehalt, 1996
AS Jahr
FROM Mitarbeiter
ORDER BY Mitarbeiter.Name

SELECT COUNT (*)  FROM Mitarbeiter

SELECT  AbtNr, COUNT (PersNr) AS Mitarbeiter
FROM Mitarbeiter
GROUP BY AbtNr

SELECT AbtNr,
       COUNT(PersNr) AS Mitarbeiter,
       MAX(Gehalt) AS [max Gehalt]
FROM Mitarbeiter
GROUP BY AbtNr

SELECT AbtNr, COUNT (PersNr) AS Mitarbeiter, SUM(Gehalt)
AS [max Gehalt]
FROM Mitarbeiter
WHERE AbtNr >= 20
GROUP BY AbtNr
HAVING SUM(Gehalt) >= 50000
```

d) zu geschachtelten Abfragen

```
SELECT Name, Beruf, AbtNr FROM Mitarbeiter WHERE AbtNr=
21 AND Beruf IN
  (SELECT DISTINCT Beruf FROM Mitarbeiter WHERE AbtNr =
     (SELECT AbtNr FROM Abteilung WHERE AbtName =
"Produktion"))
```

Aufgaben zum Join

Aufgabe 1

Was bewirken die folgenden SQL-Befehle bei Verwendung der in der Aufgabe 2 (Aufgaben zu SQL-Abfragen) verwendeten Datenbank?

```
SELECT Name, AbtName
   FROM Abteilung, Mitarbeiter
   WHERE Abteilung.AbtNr = Mitarbeiter.AbtNr

SELECT AbtName, Name
   FROM Abteilung INNER JOIN Mitarbeiter
        ON Abteilung.AbtNr = Mitarbeiter.AbtNr

SELECT AbtName, Name
   FROM Abteilung INNER JOIN Mitarbeiter
        ON Abteilung.AbtNr = Mitarbeiter.AbtNr
   WHERE AbtName = "Entwicklung"
```

Aufgabe 2

Beim INNER JOIN werden nur diejenigen Datensätze berücksichtigt, für die bezüglich der Gleichheitsbeziehung entsprechende Partner in den beiden beteiligten Tabellen existieren. Beim OUTER JOIN werden auch diejenigen Datensätze berücksichtigt, die keinen entsprechenden Partner haben - in der Ergebnistabelle steht dann NULL.

Gegeben sind die beiden Tabellen X und Y:

Spalte1	Spalte2
1	A
2	B
3	C
4	D

Spalte1	Spalte2
2	U
3	V
5	W

Erstellen Sie die Ergebnistabelle für

1. einen INNER JOIN
2. einen vollständigen OUTER JOIN
3. einen LEFT JOIN, d.h. nur die linke Tabelle X ist am OUTER JOIN beteiligt.

Aufgaben zu Aktionsabfragen

Aufgabe 1

Was bewirken die folgenden SQL-Befehle bei Verwendung der in Aufgabe 2 (Aufgaben zu SQL-Abfragen) dargestellten Datenbank? Testen Sie die Wirkung aus!

```
INSERT INTO Mitarbeiter
VALUES (500, "Walkes", "Otto", "Techniker", 4100, 20,
NULL)

INSERT INTO Mitarbeiter (PersNr, Name, AbtNr, Beruf)
VALUES (501, "Didi", 20, "Techniker")

INSERT INTO Mitarbeiter ( PersNr, Name, AbtNr, ProjektNr
)
SELECT 502, "ALF", AbtNr, NULL
      FROM Abteilung
      WHERE AbtName = "Produktion"

UPDATE Mitarbeiter SET Gehalt = Gehalt + 150  WHERE
AbtNr IN
    (SELECT AbtNr FROM Abteilung  WHERE AbtLeiter IN
        (SELECT PersNr FROM Mitarbeiter WHERE Name =
"Schubert"))

DELETE FROM Mitarbeiter WHERE PersNr = 501
```

Aufgabe 2

Die folgenden Fragen beziehen sich wieder auf die in Aufgabe 2 dargestellte Datenbank.

1. Welche Mitarbeiter arbeiten in der Abteilung 21? Gefragt sind Abteilungsnummer**,** Personalnummer und Name.
2. Welche Mitarbeiter der Abteilung 21 haben den Beruf Techniker? Gefragt sind Abteilungsnummer, Personalnummer, Name und Beruf.
3. In welchen Abteilungen arbeiten die Mitarbeiter mit den Namen Sturm, Frey, Winter und Fischer? Gefragt sind Abteilungsnummer, Personalnummer und Name.
4. Welche Mitarbeiter in der Abteilung 20 sind Laborant oder Techniker? Gefragt sind Abteilungsnummer, Personalnummer, Name und Beruf.
5. Welche Mitarbeiter der Abteilung 20 verdienen zwischen 3000 und 4000 Euro? Gefragt sind Abteilungsnummer, Personalnummer, Name und Gehalt.
6. Welche Mitarbeiter arbeiten für irgendein Projekt? Gefragt sind Personalnummer, Name und Projektnummer.

7. Welche Mitarbeiter haben einen Namen, der mit H beginnt? Gefragt sind Personalnummer und Name.
8. Welche verschiedenen Berufe werden in der Abteilung 22 ausgeübt
9. Geben Sie eine Liste der Mitarbeiter aus, die in Abteilung 20 arbeiten. Die Liste soll nach Namen und Vornamen sortiert sein. Gefragt sind Abteilungsnummer, Personalnummer, Name und Vorname.
10. Geben Sie eine nach Projektnummer und Personalnummer sortierte Liste der Mitarbeiter aus, die für irgendein Projekt arbeiten. Gefragt sind Projektnummer, Personalnummer und Name.
11. Wie hoch ist das höchste Gehalt in Abteilung 21?
12. Wie hoch ist jeweils das höchste Gehalt in den einzelnen Abteilungen?
13. Wie hoch ist jeweils das niedrigste und das höchste Gehalt der Abteilungen, die mehr als 10 Mitarbeiter haben?
14. Welche Mitarbeiter haben ein Gehalt, das mindestens 3500 Euro höher ist als das niedrigste Gehalt aller Mitarbeiter? Gefragt sind Personalnummer, Name und Gehalt.
15. Welche Mitarbeiter sind in der Abteilung Informatik beschäftigt? Gefragt sind bei Join Personalnummer, Name und Abteilungsname; bei Unterabfrage Personalnummer, Name und Abteilungsnummer.
16. Welches Gehalt hat die Leitung der Abteilung Recht?
17. Welche Mitarbeiter der Produktionsabteilung arbeiten mit am Projekt Wirkstoff ABC? Gefragt sind bei Join Abteilungsname, Projektname, Name und Personalnummer; bei Unterabfrage Abteilungsnummer, Projektnummer, Name und Personalnummer.
18. Welche Mitarbeiter der Abteilung 20 haben einen Vornamen, der auch in der Abteilung 22 vorkommt? Gefragt sind Personalnummer, Name und Vorname.
19. In welchen Projekten arbeiten Mitarbeiter, die einer Abteilung angehören, die vom Mitarbeiter mit der Personalnummer 28 geleitet wird ? Gefragt sind Projektnummer und Projektname. Die Aufgabe soll durch eine mehrfach geschachtelte Abfrage gelöst werden

Lösungsheft

Kapitel 5
–
Abfragen mit SQL

Aufgaben

Aufgaben zu SQL-Abfragen

Aufgabe 1

Gegeben ist eine Datenbank mit folgenden Relationen:

```
LIEFERER(LiefererNr,LieferName,Strasse,Plz,Ort,Rabatt)
ARTIKEL(ArtikelNr,Artikel,Preis,GruppenNr,Meldebestand,
            Bestand)
ARTIKELLIEFERER(ArtikelNr,LiefererNr,BestellNr,
                  Angebotspreis)
WARENGRUPPE(GruppenNr,Gruppe)
```

a) Was bewirkt folgende Anweisung?

```
SELECT DISTINCTROW ArtikelLieferer.LieferNr,
            ArtikelLieferer.ArtikelNr,Lieferer.LieferName
FROM Lieferer INNER JOIN ArtikelLieferer
ON Lieferer.LieferNr=ArtikelLieferer.LieferNr
WHERE ArtikelLieferer.ArtikelNr=1616;
```

Es werden die beiden Tabellen LIEFERER und ARTIKELLIEFERER durch einen Join verbunden, wobei das Auswahlkriterium dieselbe Artikelnummer ist. Aus der entstandenen Tabelle werden die Artikeldaten derjenigen Zeilen (Datensätze) ausgegeben, welche die Artikelnummer 1616 besitzen.

```
SELECT DISTINCTROW Lieferer.LieferName AS Lieferant,
Lieferer.Strasse, Lieferer.OrtOrt
FROM LIEFERER
ORDER BY LIEFERER.ORT, Lieferer.LieferName
```

bewirkt eine Ausgabe in der Form:

Lieferant	Ort	Strasse
Müller&Co	Wiesbaden	Mainzer Str. 678
Testiege P.	Mainz	Im Rad 3
...	...	...

b) Eine Lieferliste ist zu erstellen, die nach Lieferer-Ort geordnet ist!

c) Was bewirkt die folgende Aktionsabfrage?

```
UPDATE DISTINCTROW ArtikelLieferer
SET ArtikelLieferer.Angebotspreis=Angebotspreis*0,97
WHERE ArtikelLieferer.LiefererNr = 12345;
```

In der Tabelle ARTIKELLIEFERER wird ein Update an denjenigen Zeilen (Datensätzen) vorgenommen, welche die Lieferernummer 12345 besitzen. Der Angebotspreis wird auf 97% reduziert.

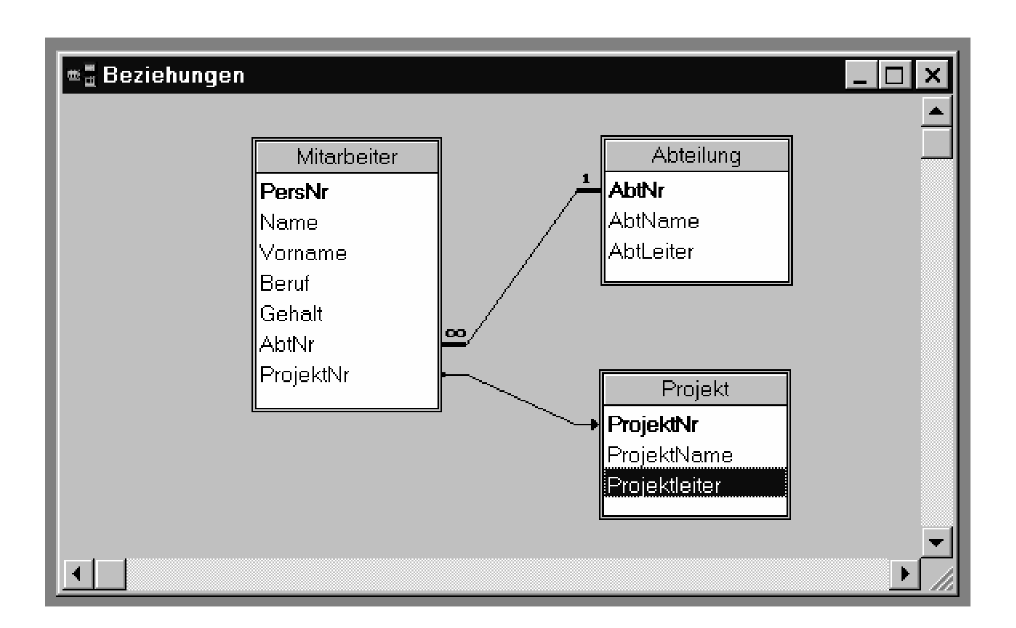
d) Welche Lieferanten gewähren einen Rabatt, der niedriger ist als der durchschnittliche Rabattsatz?

Die Abfrage lässt sich mit einer Unterabfrage bewältigen:

```
SELECT Lieferer.LieferName,Lieferer.Rabatt
FROM Lieferer
WHERE Lieferer.Rabatt < ( SELECT AVG(Rabatt)FROM Lieferer);
```

Aufgabe 2

Gegeben ist die folgende Datenbank:



In einer Firma gibt es Mitarbeiter. Jeder Mitarbeiter ist eindeutig einer Abteilung zugeordnet. Manche Mitarbeiter sind an einem Projekt beteiligt. Ein Mitarbeiter darf nur an einem Projekt arbeiten. Jede Abteilung hat einen Abteilungsleiter und jedes Projekt hat einen Projektleiter. Projektleiter und Abteilungsleiter sind natürlich auch Mitarbeiter.

Testen Sie die folgenden SQL-Befehle nach Erstellen der Datenbank und beschreiben Sie ihre Wirkung:

a) zu Projektion und Selektion

```
SELECT *
      FROM Mitarbeiter
      WHERE PersNr = 28

SELECT PersNr, Name, Vorname
      FROM Mitarbeiter
      WHERE AbtNr = 20
      ORDER BY Name, Vorname
```

Alle Daten der Tabelle MITARBEITER mit Personalnummer 28, d.h. des betreffenden Mitarbeiters Personalnummer, Name und Vorname der Mitarbeiter in Abteilung 20 sortiert nach Name

b) zu logischen Verknüpfungen und Wildcards

```
SELECT PersNr, Name, Vorname
     FROM Mitarbeiter
     WHERE Name IN ("Fritsch", "Schuster","Roller")

SELECT PersNr, Name, Vorname, Gehalt
     FROM Mitarbeiter
     WHERE Gehalt BETWEEN 2000 AND 3000

SELECT PersNr, Name, Vorname
     FROM Mitarbeiter
     WHERE Name LIKE  "Me*"

SELECT PersNr, Name, Vorname
     FROM Mitarbeiter
     WHERE ProjektNr is Null
```

- Mitarbeiterdaten Personalnummer, Name und Vorname derjenigen Mitarbeiter, die Fritsch, Schuster oder Roller heißen.

- Analog die Daten derjenigen Mitarbeiter, deren Gehalt zwischen 2000 und 3000 DM liegt.

- Analog die Daten derjenigen Mitarbeiter, deren Name mit „Me“ beginnt.

- Analog die Daten derjenigen Mitarbeiter, die in keinem Projekt mitarbeiten.

c) zur Ausgabe und zur Gruppierung

```
SELECT Mitarbeiter.PersNr, Mitarbeiter.Name AS [Name des Mitarbeiters], Mitarbeiter.Gehalt AS Monatsgehalt, 1996 AS Jahr
FROM Mitarbeiter
ORDER BY Mitarbeiter.Name


SELECT COUNT (*)  FROM Mitarbeiter


SELECT  AbtNr, COUNT (PersNr) AS Mitarbeiter
FROM Mitarbeiter
GROUP BY AbtNr


SELECT AbtNr,
       COUNT(PersNr) AS Mitarbeiter,
       MAX(Gehalt) AS [max Gehalt]
FROM Mitarbeiter
GROUP BY AbtNr


SELECT AbtNr, COUNT (PersNr) AS Mitarbeiter, SUM(Gehalt) AS [max Gehalt]
FROM Mitarbeiter
WHERE AbtNr >= 20
GROUP BY AbtNr
HAVING SUM(Gehalt) >= 50000
```

- Es werden Spaltenüberschriften erzeugt für die Spalten „Name des Mitarbeiters“ und „Monatsgehalt“; in der Spalte „Jahr“ steht die Konstante 1996, obwohl ein Attribut JAHR fehlt.

- Es wird die Zahl aller Mitarbeiter ausgegeben.

- Es wird gruppiert nach Abteilungen die Zahl der Mitarbeiter und das maximale Gehalt ermittelt und mit Spaltenüberschrift ausgegeben.

- Es erfolgt zuerst eine Selektion anhand der WHERE-Klausel, d.h. für Abteilungsnummer >=20 und anschließend wird gruppiert nach Abteilungen entsprechend der HAVING-Klausel, d.h. mit einer Gehaltssumme >=50000.

d) zu geschachtelten Abfragen

```
SELECT Name, Beruf, AbtNr FROM Mitarbeiter WHERE AbtNr= 21 AND
Beruf IN
  (SELECT DISTINCT Beruf FROM Mitarbeiter WHERE AbtNr =
     (SELECT AbtNr FROM Abteilung WHERE AbtName = "Produktion"))
```

Alle Mitarbeiter der Abteilung 21 werden ausgegeben, die einen Beruf besitzen, der auch in der Abteilung „Produktion“ vorkommt.

Aufgaben zum Join

Aufgabe 1

Was bewirken die folgenden SQL-Befehle bei Verwendung der in der Aufgabe 2 (Aufgaben zu SQL-Abfragen) verwendeten Datenbank?

```
SELECT Name, AbtName
   FROM Abteilung, Mitarbeiter
   WHERE Abteilung.AbtNr = Mitarbeiter.AbtNr

SELECT AbtName, Name
   FROM Abteilung INNER JOIN Mitarbeiter
        ON Abteilung.AbtNr = Mitarbeiter.AbtNr

SELECT AbtName, Name
   FROM Abteilung INNER JOIN Mitarbeiter
        ON Abteilung.AbtNr = Mitarbeiter.AbtNr
   WHERE AbtName = "Entwicklung"
```

Die beiden ersten SQL-Befehle liefern dasselbe Ergebnis, die erste Schreibweise entspricht dem SQL-Standard, die zweite der Access-Konvention. Es werden Name und Abteilungsname aller Mitarbeiter ausgegeben; die WHERE-Klausel unterdrückt sinnlose Kombinationen bei der Bildung des Join.
Der dritte SQL-Befehl schränkt diese Ausgabe auf die Abteilung „Entwicklung“ ein.

Aufgabe 2

Beim INNER JOIN werden nur diejenigen Datensätze berücksichtigt, für die bezüglich der Gleichheitsbeziehung entsprechende Partner in den beiden beteiligten Tabellen existieren. Beim OUTER JOIN werden auch diejenigen Datensätze berücksichtigt, die keinen entsprechenden Partner haben - in der Ergebnistabelle steht dann NULL.

Gegeben sind die beiden Tabellen X und Y:

Spalte1	Spalte2
1	A
2	B
3	C
4	D

Spalte1	Spalte2
2	U
3	V
5	W

Erstellen Sie die Ergebnistabelle für

1. einen INNER JOIN
2. einen vollständigen OUTER JOIN
3. einen LEFT JOIN, d.h. nur die linke Tabelle X ist am OUTER JOIN beteiligt.

In Access wird über die Schaltfläche Verknüpfungstyp festgelegt, ob beim Join der Equi-Join (Inner Join, Option 1) gewählt wird oder eine der beiden Inklusionsverknüfungen Outer-Join (Links-Inklusion, Option 2 bzw. Rechts-Inklusion, Option 3).

1) Beinhaltet nur die Datensätze, bei denen die Inhalte der verknüpften Felder beider Tabellen gleiche Werte haben:

X.Spalte1	**X.Spalte2**	**Y.Spalte2**
2	B	U
3	C	V

2) Ein vollständiger Outer-Join beinhaltet sowohl die Links- als auch die Rechts-Inklusion; es müssen also die Datensätze beider Tabellen vorkommen:

X.Spalte1	**X.Spalte2**	**Y.Spalte1**	**Y.Spalte2**
1	A	NULL	NULL
2	B	2	U
3	C	3	V
4	D	NULL	NULL
NULL	NULL	5	W

3) Bei der Links-Inklusion sind nur alle Datensätze der Tabelle X beteiligt:

X.Spalte1	**X.Spalte2**	**Y.Spalte1**	**Y.Spalte2**
1	A	NULL	NULL
2	B	2	U
3	C	3	V
4	D	NULL	NULL

Aufgaben zu Aktionsabfragen

Aufgabe 1

Was bewirken die folgenden SQL-Befehle bei Verwendung der in Aufgabe 2 (Aufgaben zu SQL-Abfragen) dargestellten Datenbank? Testen Sie die Wirkung aus!

```
INSERT INTO Mitarbeiter
VALUES (500, "Walkes", "Otto", "Techniker", 4100, 20, NULL)

INSERT INTO Mitarbeiter (PersNr, Name, AbtNr, Beruf)
VALUES (501, "Didi", 20, "Techniker")

INSERT INTO Mitarbeiter ( PersNr, Name, AbtNr, ProjektNr )
SELECT 502, "ALF", AbtNr, NULL
      FROM Abteilung
      WHERE AbtName = "Produktion"

UPDATE Mitarbeiter SET Gehalt = Gehalt + 150  WHERE AbtNr IN
    (SELECT AbtNr FROM Abteilung  WHERE AbtLeiter IN
        (SELECT PersNr FROM Mitarbeiter WHERE Name = "Schubert"))

DELETE FROM Mitarbeiter WHERE PersNr = 501
```

- Es handelt sich um Aktionsabfragen, welche die Datenbank in folgender Weise verändern:

- Es wird ein neuer Datensatz in die Tabelle MITARBEITER mit den angegebenen Werten eingefügt.

- Analog wird ein neuer Datensatz eingefügt, wobei die Reihenfolge der Daten nicht der Struktur der Tabelle entspricht.
Die Spezifizierung des neuen Datensatzes erfolgt diesmal nicht mit VALUES, sondern mittels SELECT anhand der Tabelle ABTEILUNG.

- Es wird das Gehalt aller Mitarbeiter um 150 erhöht, die in einer Abteilung tätig sind, deren Leiter „Schubert“ heißt.

- Es wird der Mitarbeiter mit der Personalnummer 501 aus der Tabelle entfernt.

Aufgabe 2

Die folgenden Fragen beziehen sich wieder auf die in Aufgabe 2 dargestellte Datenbank.

1. Welche Mitarbeiter arbeiten in der Abteilung 21? Gefragt sind Abteilungsnummer, Personalnummer und Name.
2. Welche Mitarbeiter der Abteilung 21 haben den Beruf Techniker? Gefragt sind Abteilungsnummer, Personalnummer, Name und Beruf.
3. In welchen Abteilungen arbeiten die Mitarbeiter mit den Namen Sturm, Frey, Winter und Fischer? Gefragt sind Abteilungsnummer, Personalnummer und Name.
4. Welche Mitarbeiter in der Abteilung 20 sind Laborant oder Techniker? Gefragt sind Abteilungsnummer, Personalnummer, Name und Beruf.
5. Welche Mitarbeiter der Abteilung 20 verdienen zwischen 3000 und 4000 Euro? Gefragt sind Abteilungsnummer, Personalnummer, Name und Gehalt.
6. Welche Mitarbeiter arbeiten für irgendein Projekt? Gefragt sind Personalnummer, Name und Projektnummer.
7. Welche Mitarbeiter haben einen Namen, der mit H beginnt? Gefragt sind Personalnummer und Name.
8. Welche verschiedenen Berufe werden in der Abteilung 22 ausgeübt
9. Geben Sie eine Liste der Mitarbeiter aus, die in Abteilung 20 arbeiten. Die Liste soll nach Namen und Vornamen sortiert sein. Gefragt sind Abteilungsnummer, Personalnummer, Name und Vorname.
10. Geben Sie eine nach Projektnummer und Personalnummer sortierte Liste der Mitarbeiter aus, die für irgendein Projekt arbeiten. Gefragt sind Projektnummer, Personalnummer und Name.
11. Wie hoch ist das höchste Gehalt in Abteilung 21?
12. Wie hoch ist jeweils das höchste Gehalt in den einzelnen Abteilungen?
13. Wie hoch ist jeweils das niedrigste und das höchste Gehalt der Abteilungen, die mehr als 10 Mitarbeiter haben?
14. Welche Mitarbeiter haben ein Gehalt, das mindestens 3500 Euro höher ist als das niedrigste Gehalt aller Mitarbeiter? Gefragt sind Personalnummer, Name und Gehalt.
15. Welche Mitarbeiter sind in der Abteilung Informatik beschäftigt? Gefragt sind bei Join Personalnummer, Name und Abteilungsname; bei Unterabfrage Personalnummer, Name und Abteilungsnummer.
16. Welches Gehalt hat die Leitung der Abteilung Recht?
17. Welche Mitarbeiter der Produktionsabteilung arbeiten mit am Projekt Wirkstoff ABC? Gefragt sind bei Join Abteilungsname, Projektname, Name und Personalnummer; bei Unterabfrage Abteilungsnummer, Projektnummer, Name und Personalnummer.
18. Welche Mitarbeiter der Abteilung 20 haben einen Vornamen, der auch in der Abteilung 22 vorkommt? Gefragt sind Personalnummer, Name und Vorname.
19. In welchen Projekten arbeiten Mitarbeiter, die einer Abteilung angehören, die vom Mitarbeiter mit der Personalnummer 28 geleitet wird ? Gefragt sind Projektnummer und Projektname. Die Aufgabe soll durch eine mehrfach geschachtelte Abfrage gelöst werden

```
1. SELECT AbtNr, PersNr, Name
   FROM Mitarbeiter
   WHERE AbtNr = 21

2. SELECT AbtNr, PersNr, Name, Beruf
   FROM Mitarbeiter
   WHERE AbtNr = 21 AND Beruf = "Techniker"


3. SELECT AbtNr, PersNr, Name
   FROM Mitarbeiter
   WHERE Name IN ("Sturm", "Frey", "Winter",
   "Fischer")

4. SELECT AbtNr, PersNr, Name, Beruf
   FROM Mitarbeiter
   WHERE AbtNr = 20 AND Beruf IN ("Laborant",
   "Techniker")


5. SELECT AbtNr, PersNr, Name, Gehalt
   FROM Mitarbeiter
   WHERE AbtNr = 20 AND Gehalt BETWEEN 3000 AND 4000

6. SELECT PersNr, Name, ProjektNr
   FROM Mitarbeiter
   WHERE ProjektNr IS NOT NULL


7. SELECT PersNr, Name
   FROM Mitarbeiter  WHERE Name  LIKE "H*"

8. SELECT DISTINCT Beruf
   FROM Mitarbeiter
   WHERE AbtNr = 22


9. SELECT AbtNr, PersNr, Name, Vorname
   FROM Mitarbeiter
   WHERE AbtNr = 20
   ORDER BY Name, Vorname

10.     SELECT ProjektNr, PersNr, Name
   FROM Mitarbeiter
   WHERE ProjektNr IS NOT NULL
   ORDER BY ProjektNr, PersNr


11.     SELECT Max (Gehalt)
   FROM Mitarbeiter
   WHERE AbtNr = 21
```

```
12.     SELECT AbtNr, Max (Gehalt)
   FROM Mitarbeiter
   GROUP BY AbtNr


13.     SELECT AbtNr, Min(Gehalt), Max (Gehalt)
   FROM Mitarbeiter
   GROUP BY AbtNr
   HAVING COUNT(PersNr) > 10

   nicht möglich ist folgende Konstruktion:

   SELECT AbtNr, Min(Gehalt), Max (Gehalt)
   FROM Mitarbeiter
   WHERE COUNT(PersNr) > 10
   GROUP BY AbtNr

14.     SELECT PersNr, Name, Gehalt
   FROM Mitarbeiter
   WHERE Gehalt >= (SELECT MIN(Gehalt) + 3500 FROM
   Mitarbeiter)

15.     SELECT PersNr, Name, AbtName
   FROM Mitarbeiter, Abteilung
   WHERE Abteilung.AbtNr = Mitarbeiter.AbtNr.
   AND  AbtName = "Informatik"

   eine weitere Lösung mit geschachteltem SELECT wäre:


   SELECT PersNr, Name, AbtNr
   FROM Mitarbeiter
   WHERE AbtNr =  (SELECT AbtNr FROM Abteilung WHERE
   AbtName = "Informatik")


16.     SELECT Gehalt
   FROM Mitarbeiter, Abteilung
   WHERE AbtName = "Recht" AND PersNr = AbtLeiter

   eine zweite Lösung ist:

   SELECT Gehalt
   FROM Mitarbeiter
   WHERE PersNr = (SELECT AbtLeiter FROM Abteilung
   WHERE AbtName = "Recht")


17.     SELECT PersNr, Name, AbtName, ProjektName
   FROM Mitarbeiter, Abteilung, Projekt
   WHERE Abteilung.AbtNr = Mitarbeiter.AbtNr
     AND AbtName = "Produktion"
     AND Mitarbeiter.ProjektNr = Projekt.ProjektNr
     AND Projektname = "Wirkstoff ABC"
```

```
eine zweite Lösung ist:
SELECT AbtNr, PersNr, Name, ProjektNr
FROM Mitarbeiter
WHERE AbtNr =  (SELECT AbtNr FROM Abteilung WHERE
AbtName = "PRODUKTION")
AND ProjektNr = (SELECT ProjektNr FROM Projekt
WHERE ProjektName = "Wirkstoff ABC")
```

18. Die Aufgabe ist ein Beispiel für Schnittmengen

```
SELECT PersNr, Name, Vorname
FROM Mitarbeiter
WHERE AbtNr = 20 AND
Vorname IN (SELECT Vorname FROM Mitarbeiter WHERE
AbtNr =22)
```

19.

```
SELECT ProjektNr, ProjektName
FROM Projekt
WHERE EXISTS
  (SELECT PersNr
  FROM Mitarbeiter
  WHERE Mitarbeiter.ProjektNr = Projekt.ProjektNr
  AND AbtNr IN
  SELECT AbtNr FROM Abteilung WHERE AbtLeiter =28))
..ORDER BY ProjektNr
```

6 Rechtliche Aspekte

Die Arbeit mit Datenbanken berührt an vielen Stellen rechtliche Probleme, da die verarbeiteten Daten und die hierzu eingesetzten Werkzeuge nicht von jedem Nutzer nach eigenem Belieben verwendet werden können.

Beispiel 1:

Zwei ehemalige Schülerinnen, die vor fünf Jahren Abitur gemacht haben, kommen ins Sekretariat der Schule und bitten um eine Adressenliste ihres Jahrgangs, um eine Wiedersehens-Party zu feiern. Die Sekretärin weigert sich mit Hinweis auf das Datenschutzgesetz, die Adressen herauszugeben. Die Ehemaligen staunen nicht schlecht, dass gleichzeitig ein Polizeibeamter die Liste einer Klasse der Jahrgangsstufe 11 erhält, die am letzten Samstag auf dem Schulgelände eine Party feierte, wobei erheblicher Sachschaden am Schulgebäude entstand.

Beispiel 2:

Auf dem Computer des Schulsekretariats werden die Personaldaten der Schüler wie auch Würdigungsberichte des Schulleiters über Lehrerinnen und Lehrer gespeichert. Der Personalrat der Lehrer fordert, dass diese Daten nicht ungeschützt allen Sekretariatsmitarbeiterinnen zugänglich sind. Der Schulträger installiert darauf ein Schutzsystem, das sich in das Betriebssystem einklinkt und jeden Zugang über Passwörter schützt. Die Daten werden verschlüsselt gespeichert. Nach einem Diebstahl des Rechners sind allerdings die Sicherungskopien wertlos, da sie nicht ohne Originalrechner gelesen werden können.

Beispiel 3:

Die Schule beschafft für den Unterricht eine Lizenz für ein Textverarbeitungssystem, das auf einem frei zugänglichen Rechner installiert wird. Kundige Schüler und Lehrer kopieren sich die Software illegal vom Rechner. Nachdem ein bekannter Software-Konzern die Schulen schriftlich auf die Strafbarkeit von Raubkopien aufmerksam gemacht hat, wird der Rechner in das Schulnetz integriert. Damit sind Kopien vom Server des Netzes, wo jetzt die Software installiert ist, für Anwender nicht mehr möglich.

Beispiel 4:

Die Schülerbibliothek der Schule verwendet ein Datenbanksystem zur Speicherung der Bücherdaten und der Ausleihvorgänge. Da die Bibliothekarin vom Förderverein der Schule bezahlt wird, fordert dieser von ihr einen Nachweis der erbrachten Arbeitsleistung und der Ausleihvorgänge. Die Schulleitung außerdem möchte von der Bibliothekarin wissen, inwieweit die verwendeten Mittel für neue Bücher auch sinnvoll eingesetzt und die gekauften Bücher auch ausgeliehen wurden. Die Datenschutzbeauftragte der Schule weist die Bibliothekarin darauf hin, dass Ausleihvorgänge nur während der Leihzeit ausleiherbezogen gespeichert werden dürfen. Anschließend seien sie zu löschen. Die Bibliothekarin weigert sich, den Bibliotheksrechner zur Arbeitskontrolle einsetzen zu lassen. Da die Ausleihvorgänge bisher nur personenbezogen gespeichert sind, muss das Programm umgeschrieben werden.

Diese vier realistischen Beispiele zeigen, dass die automatisierte Datenverarbeitung und hier vor allem die Verwendung von Dateiverwaltungssystemen und Datenbanken eine Fülle von Rechtsgütern und Rechtsvorschriften berührt, was Konsequenzen im Alltag nach sich zieht. Personen, in der Schule vor allem Schülerinnen und Schüler, Lehrerinnen und Lehrer, Sekretariatsangestellte und Mitarbeiter verschiedener Behörden sind gleichzeitig Informationsgegenstände wie Informationskonsumenten. Ihr Recht auf Informationsfreiheit kollidiert dabei oft mit den Rechten anderer, z.B. dem Recht auf Schutz der Persönlichkeit, wobei beide Rechte im Grundgesetz der Bundesrepublik verankert sind. Solche Normenkonflikte werden in einem Rechtsstaat in der Regel durch Gesetze und Verordnungen gelöst. In den oben genannten Beispielen werden Belange des Datenschutzes, der Datensicherheit und des Urheberrechts berührt, die jeweils durch entsprechende Vorschriften des Bundes und der Länder geregelt sind. In einigen Fällen ist auch das Betriebsverfassungsgesetz heranzuziehen, wenn Computer zur Steuerung und Kontrolle von Arbeitsvorgängen dienen.

Unter *Datensicherheit* versteht man die Gesamtheit aller - meist technischer und organisatorischer - Maßnahmen, die zum Schutz von Daten vor unberechtigtem Zugriff, Verfälschung oder Verlust ergriffen werden.

Unter *Datenschutz* versteht man die Gesamtheit aller - meist juristischer - Maßnahmen zum Schutze *personenbezogener Daten* vor unberechtigtem Zugriff, vor Manipulation oder Zerstörung.

Unter *Urheberrecht* sind die Vorschriften zum Schutze eines Urhebers an seinem Werk gemeint, sofern dieses schutzwürdigen Belangen entspricht, z.B. fällt auch umfangreichere Software unter das Urheberrecht.

Aufgrund dieser Begriffsbestimmung fällt es leicht, die in den Beispielen angesprochenen Tatbestände zu klassifizieren. In Beispiel 1 geht es um personenbezogene Daten, im speziellen Fall um Adresslisten. Inwieweit eine Schule solche Daten weitergeben darf, unterliegt der Datenschutz-Gesetzgebung. Ähnliches gilt in den Beispielen 2 und 4. Hier zeigt sich allerdings, dass Datenschutz und Datensicherheit nicht getrennt betrachtet werden dürfen: Entsprechende Maßnahmen zur Datensicherheit gewähren auch Datenschutz, sofern z.B. den schutzwürdigen Belangen dadurch Rechnung getragen wird, dass verschiedene Benutzergruppen nur ihre spezielle Sicht auf die Daten haben oder eine spezielle Sicht für unberechtigte Benutzer nicht möglich ist. In Beispiel 3 zeigt sich ebenfalls, dass urheberrechtlichen Belange mit geeigneten Maßnahmen zur Datensicherheit Rechnung getragen werden kann: Moderne Netzwerkumgebungen verhindern nicht nur Manipulationen an den Datenbeständen, sondern sichern auch Software gegen Raubkopieren ab.

6.1 Datenschutzregelungen

Es kommt nicht von ungefähr, dass Fragen des Datenschutzes immer an erster Stelle genannt werden, wenn es um den Einsatz von Computern bei der Speicherung und Verarbeitung von Daten geht. Dies rührt sicherlich im Wesentlichen daher, dass jeder Mensch Informationen über sich selbst als sein Eigentum und eine Weitergabe oder gar Verfälschung als Verletzung seiner Intimsphäre betrachtet.

Insofern ist das Anliegen des Datenschutzes nicht neues: Arzt- und Beichtgeheimnis, Steuergeheimnis usw. sind keine Errungenschaften unserer Zeit. Aber durch den Einsatz automatisierter Datenverarbeitungen hat das Anliegen des Schutzes personenbezogener Daten erst an Brisanz gewonnen: Nach Schätzungen ist jeder Bundesbürger durchschnittlich in etwa 200 Datenbanken gespeichert, hierzu gehören vor allem die Datenbanken der Melde- und Finanzbehörden, der Renten-, Arbeitslosen- und Krankenversicherung, der Banken und allgemeinen Versicherungen, des Kraftfahrbundesamtes in Flensburg und der Polizeibehörden, der Telekom usw. Diese Daten lassen sich schnell kopieren, über weltweite Netze übermitteln, manipulieren und löschen, ohne dass der Betroffene es überhaupt bemerken muss. Das Bundesland Hessen hat bereits 1970 das erste Datenschutzgesetz der Welt verabschiedet, in dem der Schutz von personenbezogenen Daten gesetzlich geregelt wurde. Im Jahre 1978 wurde dann das erste Bundesdatenschutzgesetz verabschiedet, inzwischen sind die Datenschutzgesetzte des Bundes und der Länder mehrfach der Entwicklung angepasst worden.

In den Datenschutzgesetzen werden meist in einem allgemeinen Teil Begriffsbestimmungen vorgenommen, dann die Rechtsgrundlagen definiert und die Rechte der Betroffenen beschrieben, die Einrichtung des Datenschutzbeauftragten geregelt und Strafvorschriften bei Verstößen gegen das Gesetz behandelt. Im folgende sind einige wesentliche Paragraphen des Hessische Datenschutzgesetzes als Beispiel für eine solche gesetzliche Regelung aufgeführt.

Hessisches Datenschutzgesetz (HDSG)

In der Fassung vom 7. Januar 1999

§1 Aufgabe

(1) Aufgabe des Gesetzes ist es, die Verarbeitung personenbezogener Daten durch die in §3 Abs.1 genannten Stellen zu regeln, um

1. das Recht des einzelnen zu schützen, selbst über die Preisgabe und Verwendung seiner Daten zu bestimmen, soweit keine Einschränkungen in diesem Gesetz oder in anderen Rechtsvorschriften zugelassen sind,

2. das auf dem Grundsatz der Gewaltenteilung beruhende verfassungsmäßige Gefüge des Staates, insbesondere der Verfassungsorgane des Landes und der Organe der kommunalen Selbstverwaltung untereinander und zueinander, vor einer Gefährdung infolge der automatisierten Datenverarbeitung zu bewahren.

(2) Aufgabe der obersten Landesbehörden, Gemeinden und Landkreise sowie der sonstigen der Aufsicht des Landes unterstehenden juristischen Personen des öffentlichen Rechts ist es, die Ausführung dieses Gesetzes sowie anderer Vorschriften über den Datenschutz jeweils für ihren Bereich sicherzustellen.

§2 Begriffsbestimmungen

(1) Personenbezogene Daten sind Einzelangaben über persönliche und sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person (Betroffener)

(2) Datenverarbeitung ist jede Verwendung gespeicherter oder zur Speicherung vorgesehener personenbezogener Daten. Im Sinne der nachfolgenden Vorschriften ist

1. Erheben das Beschaffen von Daten über den Betroffenen,

2. Speichern das Erfassen, Aufnehmen oder Aufbewahren von Daten auf einem Datenträger zum Zwecke ihrer weiteren Verarbeitung,

3. Übermitteln das Bekannt geben gespeicherter oder durch Datenverarbeitung gewonnener Daten an einen Dritten in der Weise, dass die Daten durch die Daten verarbeitende Stelle an den Dritten weitergegeben werden oder dass der Dritte zum Abruf bereitgehaltene Daten abruft,

4. Sperren das Verhindern weiterer Verarbeitung gespeicherter Daten,

5. Löschen das Unkenntlichmachen gespeicherter Daten ungeachtet der dabei angewendeten Verfahren.

(3) Datenverarbeitende Stelle ist jede der in §3 Abs.1 genannten Stellen, die Daten für sich selbst verarbeitet oder durch andere verarbeiten lässt.

(4) Empfänger ist jede Person oder Stelle, die Daten erhält.

(5) Dritter ist jede Person oder Stelle außerhalb der Daten verarbeitenden Stelle, ausgenommen der Betroffene oder diejenigen Personen und Stellen, die innerhalb des Geltungsbereichs der EG-Datenschutzrichtlinie Daten im Auftrag verarbeiten.

(6) Automatisiert ist eine Datenverarbeitung, wenn sie durch Einsatz eines gesteuerten technischen Verfahrens selbsttätig abläuft.

(7) Eine Akte ist jede der Aufgabenerfüllung dienende Unterlage, die nicht Teil der automatisierten Datenverarbeitung ist.

(8) Soweit andere landesrechtliche Vorschriften den Dateibegriff verwenden, ist Datei

1. eine Sammlung von Daten, die durch automatisierte Verfahren ausgewertet werden kann (automatisierte Datei), oder

2. eine gleichartig aufgebaute Sammlung von Daten, die nach bestimmten Merkmalen geordnet und ausgewertet werden kann (nicht-automatisierte Datei).

§3 Anwendungsbereich

(1) Dieses Gesetz gilt für Behörden und sonstige öffentliche Stellen des Landes, der Gemeinden und Landkreise sowie der sonstigen der Aufsicht des Landes unterstehenden juristischen Personen des öffentlichen Rechts und für deren Vereinigungen ungeachtet ihrer Rechtsform. Dieses Gesetz gilt auch für nicht-öffentliche Stellen, soweit sie hoheitliche Aufgaben unter Aufsicht der in Satz1 genannten Stellen wahrnehmen. .....

§ 7 Zulässigkeit der Datenverarbeitung

(1) Die Verarbeitung personenbezogener Daten ist nur zulässig, wenn

1. eine diesem Gesetz vorgehende Rechtsvorschrift sie vorsieht oder zwingend voraussetzt,

2. dieses Gesetz sie zulässt oder

3. der Betroffene ohne jeden Zweifel eingewilligt hat.

(2) Die Einwilligung bedarf der Schriftform, soweit nicht wegen besonderer Umstände eine andere Form angemessen ist. Sie muss sich im Falle einer Datenverarbeitung nach Abs.4 ausdrücklich auch auf die dort genannten Daten beziehen. Wird die Einwilligung zusammen mit anderen Erklärungen schriftlich erteilt, ist der Betroffene hierauf schriftlich besonders hinzuweisen. Der Betroffene ist in geeigneter Weise über die Bedeutung der Einwilligung, insbesondere über den Verwendungszweck der Daten, aufzuklären. Die Aufklärungspflicht umfasst bei beabsichtigten Übermittlungen auch den Empfänger der Daten. Der Betroffene ist unter Darlegung der Rechtsfolgen darauf hinzuweisen, dass er die Einwilligung verweigern und jederzeit mit Wirkung für die Zukunft widerrufen kann.

(3) Unzulässig ist eine zu rechtlichen Folgen oder erheblichen Beeinträchtigungen für den Betroffenen führende Entscheidung, wenn sie auf einer Bewertung einzelner Merkmale seiner Person beruht, die ausschließlich durch eine automatisierte Verarbeitung seiner Daten erstellt wurde. Eine Entscheidung nach Satz1 kann durch Gesetz zugelassen werden, das die Wahrung der berechtigten Interessen des Betroffenen sicherstellt

(4) Soweit nicht eine Rechtsvorschrift die Verarbeitung personenbezogener Daten über die rassische und ethnische Herkunft, politische Meinungen, religiöse oder philosophische Überzeugungen, die Gewerkschaftszugehörigkeit, die Gesundheit oder das Sexualleben vorsieht oder zwingend voraussetzt, darf eine Verarbeitung nur nach §§33 bis 35 und 39 erfolgen. Im übrigen ist eine Verarbeitung aufgrund dieses Gesetzes nur zulässig, wenn sie ausschließlich im Interesse des Betroffenen liegt und der Hessische Datenschutzbeauftragte vorab gehört worden ist. ....

§8 Rechte der Betroffenen

(1) Jeder hat nach Maßgabe dieses Gesetzes ein Recht auf

1. Auskunft und Benachrichtigung über die zu seiner Person gespeicherten Daten (§18),

2. Überprüfung der rechtmäßigen Verarbeitung seiner Daten aufgrund von ihm vorgebrachter besonderer persönlicher Gründe (§7 Abs.5),

3. Einsicht in das Verfahrensverzeichnis (§6 Abs.2),

4. Berichtigung, Sperrung oder Löschung der zu seiner Person gespeicherten Daten (§19),

5. Schadensersatz (§20),

6. Anrufung des Datenschutzbeauftragten (§§28 und 37 Abs.2).

(2) Wenn eine in §3 Abs.1 genannte Stelle für die Gewährung einer Leistung, das Erkennen einer Person oder für einen anderen Zweck einen Datenträger herausgibt, auf dem personenbezogene Daten des Inhabers automatisiert verarbeitet werden, etwa in Form einer Chipkarte, dann hat sie sicherzustellen, dass er dies erkennen und seine ihm nach Abs.1 Nr.1 bis 5 zustehenden Rechte ohne unvertretbaren Aufwand geltend machen kann. Der Inhaber ist bei Ausgabe des Datenträgers über die ihm nach Abs.1 zustehenden Rechte sowie über die von ihm bei Verlust des Datenträgers zu treffenden Maßnahmen und über die Folgen aufzuklären

§9 Datengeheimnis

Den bei der Daten verarbeitenden Stelle oder in deren Auftrag beschäftigten Personen, die Zugang zu personenbezogenen Daten haben, ist eine Verarbeitung dieser Daten zu einem anderen als dem zur jeweiligen rechtmäßigen Aufgabenerfüllung gehörenden Zweck während und nach Beendigung ihrer Tätigkeit untersagt. Diese Personen sind über die bei ihrer Tätigkeit zu beachtenden Vorschriften über den Datenschutz zu unterrichten.

§ 10 Technische und organisatorische Maßnahmen

(1) Die Daten verarbeitende oder in ihrem Auftrag tätige Stelle hat die technischen und organisatorischen Maßnahmen zu treffen, die nach Abs.2 und 3 erforderlich sind, um die Ausführung dieses Gesetzes sowie anderer Vorschriften über den Datenschutz zu gewährleisten. Erforderlich sind diese Maßnahmen, soweit der damit verbundene Aufwand unter Berücksichtigung der Art der personenbezogenen Daten und ihrer Verarbeitung zum Schutz des in §1 Abs.1 Nr.1 genannten Rechts angemessen ist.

(2) Werden personenbezogene Daten automatisiert verarbeitet, ist das Verfahren auszuwählen oder zu entwickeln, welches geeignet ist, so wenig personenbezogene Daten zu verarbeiten, wie zur Erreichung des angestrebten Zwecks erforderlich ist. Außerdem sind Maßnahmen schriftlich anzuordnen, die nach dem jeweiligen Stand der Technik und der Art des eingesetzten Verfahrens erforderlich sind, um zu gewährleisten, dass

1. Unbefugte keinen Zutritt zu Datenverarbeitungsanlagen, mit denen personenbezogene Daten verarbeitet werden, erhalten (Zutrittskontrolle),

2. Unbefugte an der Benutzung von Datenverarbeitungsanlagen und -verfahren gehindert werden (Benutzerkontrolle),

3. die zur Benutzung eines Datenverarbeitungsverfahrens Befugten ausschließlich auf die ihrer Zugriffsberechtigung unterliegenden personenbezogenen Daten zugreifen können (Zugriffskontrolle),

4. personenbezogene Daten nicht unbefugt oder nicht zufällig gespeichert, zur Kenntnis genommen, verändert, kopiert, übermittelt, gelöscht, entfernt, vernichtet oder sonst verarbeitet werden (Datenverarbeitungskontrolle),

5. es möglich ist, festzustellen, wer welche personenbezogenen Daten zu welcher Zeit verarbeitet hat und wohin sie übermittelt werden sollen oder übermittelt worden sind (Verantwortlichkeitskontrolle),

6. personenbezogene Daten, die im Auftrag verarbeitet werden, nur entsprechend den Weisungen des Auftraggebers verarbeitet werden können (Auftragskontrolle),

7. durch eine Dokumentation aller wesentlichen Verarbeitungsschritte die Überprüfbarkeit der Datenverarbeitungsanlage und des -verfahrens möglich ist (Dokumentationskontrolle),

8. die innerbehördliche oder innerbetriebliche Organisation den besonderen Anforderungen des Datenschutzes gerecht wird (Organisationskontrolle).

(3) Werden personenbezogene Daten nicht automatisiert verarbeitet, dann sind insbesondere Maßnahmen zu treffen, um den Zugriff Unbefugter bei der Bearbeitung, der Aufbewahrung, dem Transport und der Vernichtung zu verhindern.

§11 Erforderlichkeit

(1) Die Verarbeitung personenbezogener Daten ist nach Maßgabe der nachfolgenden Vorschriften zulässig, wenn sie zur rechtmäßigen Erfüllung der in der Zuständigkeit der Daten verarbeitenden Stelle liegenden Aufgaben und für den jeweils damit verbundenen Zweck erforderlich ist. Die Erforderlichkeit einer Datenübermittlung muss nur bei einer der beteiligten Stellen vorliegen.

(2) Sind personenbezogene Daten in Akten derart verbunden, dass ihre Trennung nach erforderlichen und nicht erforderlichen Daten nicht oder nur mit unverhältnismäßig großem Aufwand möglich ist, dann sind die Kenntnisnahme, die Weitergabe innerhalb der Daten verarbeitenden Stelle und die Übermittlung der Daten, die nicht zur Erfüllung der jeweiligen Aufgabe erforderlich sind, über Abs. 1 hinaus zulässig. Diese Daten unterliegen insoweit einem Verwertungsverbot.

§ 12 Erheben

(1) Personenbezogene Daten sind grundsätzlich bei dem Betroffenen mit seiner Kenntnis zu erheben. Werden Daten nicht über eine bestimmte Person, sondern über einen bestimmbaren Personenkreis, etwa durch Videoüberwachung, erhoben, dann genügt es, wenn er die seinen schutzwürdigen Belangen angemessene Möglichkeit zur Kenntnisnahme hat.

(2) Bei öffentlichen Stellen dürfen Daten im Einzelfall ohne seine Kenntnis nur erhoben werden, wenn

1. eine Rechtsvorschrift dies vorsieht, zwingend voraussetzt oder der Betroffene eingewilligt hat,

2. die Bearbeitung eines vom Betroffenen gestellten Antrags ohne Kenntnis der Daten nicht möglich ist oder Angaben des Betroffenen überprüft werden müssen; der Betroffenen ist darauf hinzuweisen, bei welchen Personen oder Stellen seine Daten erhoben werden können,

3. die Abwehr erheblicher Nachteile für das Allgemeinwohl oder von Gefahren für Leben, Gesundheit und persönliche Freiheit dies gebietet,

4. sich bei Gelegenheit der rechtmäßigen Aufgabenerfüllung Anhaltspunkte für Straftaten oder Ordnungswidrigkeiten ergeben oder

5. die Erhebung beim Betroffenen einen unverhältnismäßigen Aufwand erfordern würde und keine Anhaltspunkte dafür bestehen, dass schutzwürdige Belange des Betroffenen beeinträchtigt werden können.

(3) Beim Betroffenen und bei Dritten außerhalb des öffentlichen Bereichs dürfen Daten ohne seine Kenntnis nur erhoben werden, wenn der Schutz von Leben und Gesundheit oder die Abwehr einer erheblichen Gefährdung der natürlichen Lebensgrundlagen dies im Einzelfall gebietet oder eine Rechtsvorschrift dies vorsieht oder, soweit es sich um eine Rechtsvorschrift des Bundes handelt, zwingend voraussetzt.

§13 Zweckbindung

(1) Personenbezogene Daten dürfen grundsätzlich nur für den Zweck weiterverarbeitet werden, für den sie erhoben oder gespeichert worden sind.

(2) Sollen personenbezogene Daten zu Zwecken verarbeitet werden, für die sie nicht erhoben oder gespeichert worden sind, dann ist dies nur aus den in §12 Abs.2 und 3 genannten Gründen zulässig. Besondere Amts- oder Berufsgeheimnisse bleiben unberührt.

(3) Sind personenbezogene Daten in Akten derart verbunden, dass ihre Trennung nach verschiedenen Zwecken nicht oder nur mit unvertretbar großem Aufwand möglich ist, so tritt an die Stelle der Trennung ein Verwertungsverbot nach Maßgabe von Abs.2 für die Daten, die nicht dem Zweck der jeweiligen Verarbeitung dienen.

(4) Personenbezogene Daten, die für andere Zwecke erhoben worden sind, dürfen auch zur Ausübung von Aufsichts- und Kontrollbefugnissen sowie zu Ausbildungs- und Prüfungszwecken in dem dafür erforderlichen Umfang verwendet werden.

(5) Personenbezogene Daten, die ausschließlich zu Zwecken der Datenschutzkontrolle, der Datensicherung oder zur Sicherstellung des ordnungsgemäßen Betriebes einer Datenverarbeitungsanlage gespeichert werden, dürfen nicht für andere Zwecke verwendet

§16 Übermittlung an Personen oder Stellen außerhalb des öffentlichen Bereichs

(1) Die Übermittlung personenbezogener Daten an Personen oder Stellen außerhalb des öffentlichen Bereichs ist über §§11 und 13 hinaus zulässig, wenn der Empfänger ein berechtigtes Interesse an der Kenntnis der zu übermittelnden Daten glaubhaft macht und keine Anhaltspunkte dafür bestehen, dass schutzwürdige Belange des Betroffenen beeinträchtigt werden können.

(2) Der Empfänger darf die übermittelten Daten nur zu dem Zweck verwenden, zu dessen Erfüllung sie ihm übermittelt wurden

§ 18 Auskunft und Benachrichtigung

(1) Datenverarbeitende Stellen, die personenbezogene Daten automatisiert speichern, haben die Betroffenen von dieser Tatsache schriftlich zu benachrichtigen und dabei die Art der Daten sowie die Zweckbestimmung und die Rechtsgrundlage der Speicherung zu nennen. Die Benachrichtigung erfolgt zum Zeitpunkt der Speicherung oder im Fall einer beabsichtigten Übermittlung spätestens mit deren Durchführung. Dienen die Daten der Erstellung einer beabsichtigten Mitteilung an den Betroffenen, kann die Benachrichtigung mit dieser Mitteilung verbunden werden.

(2) Eine Pflicht zur Benachrichtigung besteht nicht, wenn

1. die Daten beim Betroffenen erhoben oder von ihm mitgeteilt worden sind,

2. die Verarbeitung der personenbezogenen Daten durch Gesetz ausdrücklich vorgesehen ist,

3. der Betroffene auf andere Weise Kenntnis von der Verarbeitung seiner Daten erlangt hat,

4. die Benachrichtigung des Betroffenen unmöglich ist oder einen unverhältnismäßigen Aufwand erfordert.

(3) Datenverarbeitende Stellen, die personenbezogene Daten automatisiert speichern, haben dem Betroffenen auf Antrag gebührenfrei Auskunft zu erteilen über

1. die zu seiner Person gespeicherten Daten

2. den Zweck und die Rechtsgrundlage der Verarbeitung sowie

3. die Herkunft der Daten und die Empfänger übermittelter Daten, soweit dies gespeichert ist.

In dem Antrag soll die Art der Daten, über die Auskunft erteilt werden soll, näher bezeichnet werden.

(4) Abs.1 und 3 gelten nicht für personenbezogene Daten, die deshalb gesperrt sind, weil sie auf Grund gesetzlicher Aufbewahrungsvorschriften nicht gelöscht werden dürfen, sowie für solche Daten, die ausschließlich zum Zwecke der Datensicherung oder Datenschutzkontrolle gespeichert werden.

(5) Sind personenbezogene Daten in Akten gespeichert, die zur Person des Betroffenen geführt werden, dann kann er bei der aktenführenden Stelle Einsicht in die von ihm bezeichneten Akten verlangen. Werden die Akten nicht zur Person des Betroffenen geführt, hat er Angaben zu machen, die das Auffinden der zu seiner Person gespeicherten Daten mit angemessenem Aufwand ermöglichen. Die Einsichtnahme ist unzulässig, wenn die Daten des Betroffenen mit Daten Dritter oder geheimhaltungsbedürftigen nicht personenbezogenen Daten derart verbunden sind, dass ihre Trennung nicht oder nur mit unverhältnismäßig großem Aufwand möglich ist. In diesem Fall ist dem Betroffenen Auskunft nach Abs.3 zu erteilen. Im Übrigen kann ihm statt Einsicht Auskunft gewährt werden.

(6) Abs.1 und 3 gelten nicht, soweit eine Abwägung ergibt, dass die dort gewährten Rechte des Betroffenen hinter dem öffentlichen Interesse an der Geheimhaltung oder einem überwiegenden Geheimhaltungsinteresse Dritter zurücktreten müssen. Die Entscheidung trifft der Leiter der verpflichteten Stelle oder dessen Stellvertreter. Werden Auskunft oder Einsicht nicht gewährt, ist der Betroffene unter Mitteilung der wesentlichen Gründe darauf hinzuweisen, dass er sich an den Hessischen Datenschutzbeauftragten wenden kann.

(7) Bei Prüfungs- und Berufungsverfahren können die in Abs. 1 bis 6 gewährten Rechte erst nach dem Verfahrensabschluss geltend gemacht werden.

§ 19 Berichtigung, Sperrung und Löschung

(1) Personenbezogene Daten sind zu berichtigen, wenn sie unrichtig sind.

(2) Personenbezogene Daten sind zu sperren, wenn

1. ihre Richtigkeit vom Betroffenen bestritten wird und sich weder die Richtigkeit noch die Unrichtigkeit feststellen lässt,

2. ihre Verarbeitung unzulässig ist und die Löschung den Betroffenen in der Verfolgung seiner Rechte beeinträchtigen würde.

In automatisierten Verfahren ist die Sperrung grundsätzlich durch technische Maßnahmen sicherzustellen; im übrigen ist ein entsprechender Vermerk anzubringen. Gesperrte Daten dürfen über die Speicherung hinaus nicht mehr verarbeitet werden, es sei denn, dass die Verarbeitung zur Behebung einer bestehenden Beweisnot oder aus sonstigen im rechtlichen Interesse eines Dritten liegenden Gründen unerlässlich ist oder der Betroffene in die Verarbeitung eingewilligt hat.

(3) Personenbezogene Daten sind unverzüglich zu löschen, sobald feststeht, dass ihre Speicherung nicht mehr erforderlich ist, um die Zwecke zu erfüllen, für die sie erhoben worden sind oder für die sie nach §13 Abs.2 und 4 weiterverarbeitet werden dürfen. Wenn bei der Speicherung nicht absehbar ist, wie lange die Daten benötigt werden, ist nach einer aufgrund der Erfahrung zu bestimmenden Frist zu prüfen, ob die Erforderlichkeit der Speicherung noch besteht. Satz1 findet keine Anwendung, wenn Grund zu der Annahme besteht, dass durch die Löschung schutzwürdige Belange des Betroffenen beeinträchtigt werden.

§ 34 Datenschutz bei Dienst- und Arbeitsverhältnissen

(1) Der Dienstherr oder Arbeitgeber darf Daten seiner Beschäftigten nur verarbeiten, wenn dies zur Eingehung, Durchführung, Beendigung oder Abwicklung des Dienst- oder Arbeitsverhältnisses oder zur Durchführung innerdienstlicher planerischer, organisatorischer, sozialer und personeller Maßnahmen erforderlich ist oder eine Rechtsvorschrift, ein Tarifvertrag oder eine Dienstvereinbarung es vorsieht. Die für das Personalaktenrecht geltenden Vorschriften des

Hessischen Beamtengesetzes sind, soweit tarifvertraglich nichts anderes geregelt ist, auf Angestellte und Arbeiter im öffentlichen Dienst entsprechend anzuwenden.

(2) Abweichend von §16 Abs.1 ist eine Übermittlung der Daten von Beschäftigten an Personen und Stellen außerhalb des öffentlichen Bereichs nur zulässig, wenn der Empfänger ein rechtliches Interesse darlegt, der Dienstverkehr es erfordert oder der Betroffene eingewilligt hat. Die Übermittlung an einen künftigen Dienstherrn oder Arbeitgeber ist nur mit Einwilligung des Betroffenen zulässig.

(3) Das Auskunftsrecht nach §18 Abs.3 umfasst auch die Art der automatisierten Auswertung der Daten des Beschäftigten. §18 Abs.6 findet keine Anwendung.

(4) Im Falle des §19 Abs.3 Satz1 sind die Daten der Beschäftigten zu löschen. Daten, die vor der Eingehung eines Dienst- oder Arbeitsverhältnisses erhoben wurden, sind unverzüglich zu löschen, sobald feststeht, dass ein Dienst- oder Arbeitsverhältnis nicht zustande kommt. Dies gilt nicht, wenn Grund zu der Annahme besteht, dass durch die Löschung schutzwürdige Belange des Betroffenen beeinträchtigt werden.

(5) Vor Einführung, Anwendung, Änderung oder Erweiterung eines automatisierten Verfahrens zur Verarbeitung von Daten der Beschäftigten hat die Dienststelle das Verfahrensverzeichnis (§6) der Personalvertretung im Rahmen des personalvertretungsrechtlichen Beteiligungsverfahrens mit dem Hinweis vorzulegen, dass sie eine Stellungnahme des Hessischen Datenschutzbeauftragten fordern kann. Macht die Personalvertretung von dieser Möglichkeit Gebrauch, beginnt die von ihr einzuhaltende Frist erst mit der Vorlage der von der Dienststellenleitung einzuholenden Stellungnahme.

(6) Daten der Beschäftigten, die im Rahmen der Durchführung der technischen und organisatorischen Maßnahmen nach §10 Abs.2 gespeichert werden, dürfen nicht zu Zwecken der Verhaltens- oder Leistungskontrolle ausgewertet werden.

Das Hessische Datenschutzgesetz regelt allein den Datenschutz bei der Verarbeitung personenbezogener Daten durch öffentliche Stellen, u.a. durch Behörden, durch Anstalten des öffentlichen Rechts (z.B. Rundfunk) oder den Landtag. Das Bundesdatenschutzgesetz (BDSG) regelt dagegen auch die Verarbeitung personenbezogener Daten bei privaten Stellen, allerdings nur in relativ allgemeiner Form. Bereichsspezifische Regelungen finden sich in anderen Gesetzen.

Wesentliche Aussagen hat 1983 hat das Bundesverfassungsgericht (BVG) aufgrund einer Klage gegen die damals geplante Volkszählung zu den Rechten der Betroffenen gemacht: Das BVG hat aus den in Artikeln 1 und 2 des Grundgesetzes (s. unten) gewährleisteten allgemeinen Persönlichkeitsrechten das *Recht auf informationelle Selbstbestimmung* abgeleitet. Danach ist jeder einzelne befugt, „grundsätzlich selbst über die Preisgabe und Verwendung seiner persönlichen Daten zu bestimmen“. Jede Verarbeitung personenbezogener Daten ist demnach ein *Grundrechtseingriff*, der nur zulässig ist, wenn der Betroffene zustimmt oder aber ein Gesetz dies eindeutig erlaubt.. Auch hier hat das BVG enge Grenzen gezogen: “Ein Zwang zur Angabe personenbezogener Daten setzt voraus, dass der Gesetzgeber den Verwendungszweck bereichsspezifisch und präzise bestimmt und dass die Angaben für diesen Zweck geeignet und erforderlich sind“.

Grundgesetz für die Bundesrepublik Deutschland

vom 23.5.1949 (Auszug)

Artikel 1

Die Würde des Menschen

Die Würde des Menschen ist unantastbar. Sie zu achten und zu schützen ist Verpflichtung aller staatlichen Gewalt.

Das Deutsche Volk bekennt sich darum zu unverletzlichen und unveräußerlichen Menschenrechten als Grundlage jeder menschlichen Gemeinschaft, des friedens und der Gerechtigkeit in der Welt.

Die nachfolgenden Grundrechte binden Gesetzgebung, vollziehende Gewalt und Rechtsprechung als unmittelbar geltendes Recht.

Artikel 2

Freiheit der Person, Recht auf Leben und körperliche Unversehrtheit

Jeder hat das Recht auf die freie Entfaltung seiner Persönlichkeit, soweit er nicht die Rechte anderer verletzt und nicht gegen die verfassungsmäßige Ordnung oder das Sittengesetz verstößt.

Jeder hat das Recht auf Leben und körperliche Unversehrtheit. Die Freiheit der Person ist unverletzlich. In diese Rechte darf nur auf Grund eines Gesetzes eingegriffen werden.

Die Datenschutzgesetzgebung hat diesem Urteils des BVG Rechnung getragen, das den Datenschutz zum Grundrecht erhoben hat. Neben den eigentlichen Datenschutzgesetzen regeln noch weitere Gesetze die Verarbeitung personenbezogener Daten. Hierzu gehören das Melderechtsrahmengesetz des Bundes und die Meldegesetze der Länder, das Bürgerliche Gesetzbuch, das Strafgesetzbuch, das Urheberrechtsgesetz, das Bundesstatistikgesetz und entsprechende Gesetze der Länder, das Betriebsverfassungsgesetz sowie weitere ca. 130 gesetzliche Vorschriften.

6.2 Datenschutz in der Schule

Die Datenschutzgesetze des Bundes und der Länder stecken den rechtlichen Rahmen ab für weitere bereichsspezifische Regelungen, wie sie vom BVG gefordert wurden. In der Schule werden seit jeher die Daten von Schülerinnen und Schülern, Erziehungsberechtigten und Unterrichtenden verarbeitet, ohne dass die Betroffenen ausdrücklich zustimmen. Das Hessische Schulgesetz von 1992 und die zugehörige Verordnung über die Verarbeitung personenbezogener Daten in Schulen von 1993 (vgl. [HKM01]) regeln sehr detailliert, welche Daten zu welchen Zwecken in den Schulen erhoben und wozu verarbeitet werden dürfen. Enthalten sind dabei Regelungen für

- den Anwendungsbereich der Datenschutzvorschriften auf Schulen in freier Trägerschaft
- die Verarbeitungsbefugnisse der Schulen, der Schulaufsichtsbehörden und der Schulträger
- die automatisierte Datenverarbeitung
- die Forschung im Schulbereich
- Ordnungsmaßnahmen (vor allem Tilgungsvorschriften)
- die Informationsrechte der Schüler und Erziehungsberechtigten
- minderjährige Schüler (außer zu Forschungszwecken können auch minderjährige Schüler der Verarbeitung ihrer Daten zustimmen, sofern sie in der Lage sind, die Tragweite ihrer Entscheidung abzuschätzen)
- Schulgesundheitspflege und schulpsychologischen Dienst
- die Datensicherheit
- Lehrerdaten
- die Bestellung des schulinternen Datenschutzbeauftragten.

Die folgenden Auszüge aus den hessischen Bestimmungen machen deutlich, dass der Datenschutz in der Schule vom Gesetzgeber für besonders wichtig erachtet wird und der Umfang der Regelungen inzwischen entsprechend angewachsen ist.

Hessisches Schulgesetz vom 17. Juni 1992 (GVBl. I S. 233), geändert durch das Erste Gesetz zur Qualitätssicherung in hessischen Schulen vom 30. Juni 1999 (GVBl. I S. 354) (Schulgesetz - HSchG -), zuletzt geändert am 2. April 2001 (§92 und §99).

§ 72 Informationsrechte der Eltern und der Schülerinnen und Schüler

(1)
Die Schülerinnen und Schüler und ihre Eltern sind in allen wichtigen Schulangelegenheiten zu informieren und zu beraten. Dazu gehören insbesondere

Aufbau und Gliederung der Schule und der Bildungsgänge,

die Übergänge zwischen den Bildungsgängen,

die Abschlüsse und Berechtigungen einschließlich der Zugänge zu den Berufen,

Grundzüge der Planung und Gestaltung des Unterrichts, Grundzüge der Unterrichtsinhalte und Unterrichtsziele sowie der Leistungsbewertung einschließlich Versetzung und Kurseinstufung.

(2)
Die Information und Beratung der Eltern erfolgen in der Regel in den Elternversammlungen, bei den Schülerinnen und Schülern in der Regel im Rahmen des Unterrichts. Mit Zustimmung der Lehrerin oder des Lehrers und mit Einverständnis der Schulleiterin oder des Schulleiters können die Eltern in der Grundstufe (Primarstufe) und in der Mittelstufe (Sekundarstufe I) den Unterricht ihrer Kinder besuchen.

(3)
Die Schulleiterin oder der Schulleiter sowie die Lehrerinnen und Lehrer sollen die Eltern und Schülerinnen und Schüler in angemessenem Umfang informieren und beraten über

die Lernentwicklung sowie das Arbeits- und Sozialverhalten der Schülerin oder des Schülers, insbesondere bei Lern- und Verhaltensstörungen,

die Leistungsbewertung einschließlich der Versetzungen und Kurseinstufungen sowie

die Wahl der Bildungsgänge.

(4)
Jugendliche, die Eltern und volljährige Schülerinnen und Schüler haben das Recht, Akten der Schule, Schulaufsichtsbehörden und des schulärztlichen Dienstes, in denen Daten über sie gespeichert sind, einzusehen. Die Einsichtnahme ist unzulässig, wenn die Daten der Betroffenen mit Daten Dritter derart verbunden sind, dass die Trennung nicht oder nur mit unverhältnismäßig großem Aufwand möglich ist. In diesem Fall ist den Betroffenen über die zu ihrer Person gespeicherten Daten Auskunft zu erteilen...

§ 82 Pädagogische Maßnahmen und Ordnungsmaßnahmen

(10)
Eintragungen und Vorgänge über Ordnungsmaßnahmen sind spätestens am Ende des zweiten Schuljahres nach der Eintragung zu löschen, sofern nicht während dieser Zeit eine erneute Ordnungsmaßnahme getroffen wurde.

§ 83 Erhebung und Verarbeitung von personenbezogenen Daten

(1)
Schulen dürfen personenbezogene Daten von Schülerinnen und Schülern, deren Eltern und

Lehrerinnen und Lehrern verarbeiten, soweit dies zur rechtmäßigen Erfüllung des Bildungs- und Erziehungsauftrags der Schule und für einen jeweils damit verbundenen Zweck oder zur Durchführung schulorganisatorischer Maßnahmen erforderlich ist.

(2)
Schulträger und Schulaufsichtsbehörden dürfen personenbezogene Daten von Schülerinnen und Schülern, deren Eltern und Lehrerinnen und Lehrern verarbeiten, soweit dies zur rechtmäßigen Erfüllung der in ihrer Zuständigkeit liegenden Aufgaben der Schulplanung, der Schulorganisation und der Schulaufsicht und einem jeweils damit verbundenen Zweck oder zur Durchführung organisatorischer Maßnahmen erforderlich ist.

(3)
Schülerinnen und Schüler, deren Eltern und Lehrerinnen und Lehrer sind verpflichtet, die erforderlichen Angaben zu machen.

(4)
Im Rahmen der Schulgesundheitspflege und des schulpsychologischen Dienstes dürfen die für die Durchführung der schulärztlichen oder schulpsychologischen Untersuchungen sowie sonderpädagogischen Überprüfungen nach § 71 erforderlichen personenbezogenen Daten verarbeitet werden. Der schulärztliche und der schulpsychologische Dienst dürfen der Schule nur das Ergebnis der Pflichtuntersuchungen übermitteln. Personenbezogene Daten über freiwillige Untersuchungen dürfen nur mit schriftlicher Einwilligung der Betroffenen übermittelt werden. Medizinische und psychologische Befunde dürfen nicht automatisiert verarbeitet werden, ausgenommen die medizinischen Befunde der für die Schulgesundheitspflege zuständigen Behörden (§ 149). Im schulpsychologischen Dienst eingesetzte Datenverarbeitungsgeräte dürfen nicht mit Datenverarbeitungsgeräten, die für andere Aufgaben benutzt werden, vernetzt werden; nähere Regelungen trifft das Kultusministerium in Richtlinien für den schulpsychologischen Dienst.

(5)
Personenbezogene Daten von Schülerinnen und Schülern, deren Eltern und Lehrerinnen und Lehrern dürfen in der Regel nur in der Schule verarbeitet werden. Die automatisierte Verarbeitung personenbezogener Daten darf nur auf schuleigenen Datenverarbeitungsgeräten erfolgen. Die Schulleiterin oder der Schulleiter kann in begründeten Ausnahmen gestatten, dass Lehrerinnen und Lehrer Daten von Schülerinnen und Schülern auf Datenverarbeitungsgeräten außerhalb der Schule verarbeiten.

(6)
Soweit in diesem Gesetz nichts anderes geregelt ist, gilt das Hessische Datenschutzgesetz in der jeweils geltenden Fassung.

(7)
Umfang und Einzelheiten der personenbezogenen Datenverarbeitung in der Schule werden durch Rechtsverordnung näher geregelt; dabei ist zu bestimmen, welche Daten unter welchen Auflagen Lehrerinnen und Lehrer außerhalb der Schule verarbeiten dürfen.

Verordnung über die Verarbeitung personenbezogener Daten

vom 30.11.1993 (Auszug)

§1 Umfang

Die personenbezogenen Daten von Schülerinnen und Schülern sowie von deren Eltern, die von den Schulen nach § 83 Abs. 1 des Schulgesetzes verarbeitet werden dürfen ergeben sich aus Anlage 1 aus dieser Verordnung....

Schulen dürfen von den Eltern Namen und Anschriften, einschließlich Telefonanschluß, sowie ggf. Daten über deren Funktion als Mitglied einer Elternvertretung und über die Erziehungsberechtigung erheben und verarbeiten (Anlage 1 Nr. 2).

Schulen dürfen
- für die Planung und Durchführung der Unterrichtsorganisation,
- für die Fortschreibung der amtlichen Schuldaten,

die in Anlage 2 enthaltenen Personalstammdaten der Lehrerinnen und Lehrer verarbeiten. Dies schließt die Verarbeitung weiterer lediglich schulorganisatorischer Daten, nicht aus. Im übrigen gelten die Regeln des § 34 HDSG.

§4 Automatisierte Verarbeitung

Werden für die nach § 1 zulässigen Verarbeitung personenbezogener Daten automatisierte Verfahren eingesetzt, dürfen die Datenverarbeitungsgeräte nicht für andere Zwecke genutzt und nicht mit Datenverarbeitungsgeräten für Unterrichtszwecke vernetzt werden.

§ 6 Datensichrerheitsmaßnahmen

Die Schule hat alle technischen und organisatorischen Maßnahmen im Sinne von § 10 HDSG durchzuführen, um sicherzustellen, daß bei allen Arten der Datenverarbeitung (§2 Abs. 2 HDSG) der Zugriff Unbefugter verhindert wird. Einzelheiten ergeben sich aus der Anlage 5.

§ 7 Löschung personenbezogener Daten in automatisierten Dateien

Die in automatisierten Dateien gespeicherten personenbezogenen Daten von Schülerinnen und Schülern, Lehrerinnen und Lehrern oder Eltern sind nach Abschluß der Aufgabe, für die sie erhoben und gespeichert wurden in der Regel spätestens zu dem Zeitpunkt zu löschen, zu dem die Schülerin oder der Schüler die Schule verläß.

Eine Ausnahme bilden Sicherungskopien der Daten, die für zulässige Auskünfte wie für das Ausstellen von Bescheinigungen von der Schule längerfristig benötigt wer den. Sie sind hinsichtlich Aufbewarungsfristen und Löschfristen den Schulakten gleichgestellt. ..

Die in der Verordnung angesprochenen Anlagen regeln zusätzlich eine Fülle von Details und geben konkrete Hinweise für die Datenübermittlung. So zählt Anlage 1 die Daten einzeln auf, die in der Schule verarbeitet werden dürfen, u.a.:

Schülerinnen und Schüler: Individualdaten (Name, Vorname, Adresse, Telefon, Geschlecht, Familienstand, Geburtsdatum, Geburtsort, Staatsangehörigkeit, Konfession),

Elterndaten (Namen, Vornamen, Adresse, Telefon, schulische Funktionen, Erziehungsberechtigung),

Schullaufbahndaten (u.a. Einschulung, Eintrittsdatum, Vorbildung, bisherige Schulen, Klasse, Klassenlehrer, Entlassungsdatum, Wahlfächer, Praktika, Beurlaubungen, Eignungsfeststellungen, Unterrichtsversäumnisse, Ordnungsmaßnahmen)

Leistungsdaten (Zeugnisnoten, Bemerkungen, Konferenzergebnisse, Benachrichtigungen)

Schulformspezifische Zusatzdaten (u.a. für die gymnasiale Oberstufe: Kurswahl, Leistungsbewertungen, Fremdsprachen, Abiturzulassung, Prüfungsfächer, Prüfer, Abiturergebnisse).

In Anlage 2 sind die Daten von Lehrerinnen und Lehrern aufgeführt, die in der Schule verarbeitet werden dürfen. Hierzu gehören: Name, Vorname, Kürzel, Geschlecht, Anschrift, Telefon, Dienstverhältnis, Lehramt, Funktion, Lehrbefähigung, Unterrichtserlaubnis, Pflichtstundensoll, erteilter Unterricht, Anrechnungen, Ermäßigungen, Sprechstunden, Freistellungen.

In Anlage 3 ist detailliert aufgeführt, welche Daten im Klassenbuch erfasst werden dürfen und welche nicht (z. B. Leistungsdaten).

Die Anlage 4 regelt die Bestellung des schulinternen Datenschutzbeauftragten durch den Schulleiter und die dafür notwendigen Voraussetzungen, die Stellung des Datenschutzbeauftragten in der Schule (dieser hat außer bei den Überwachungsaufgaben nach § 10 HDSG keine Kontroll- sondern nur Beratungsfunktion), die Aufgaben des Datenschutzbeauftragten (i.d.R. Mitwirkungsrechte bei allen Angelegenheiten des Datenschutzes und Informations- und Beratungsrechte).

In Anlage 5 sind die Maßnahmen zur Datensicherheit und zum Datenschutz nach § 10 HDSG erläutert. Hierzu gehört die Beschreibung zu Sicherheitsmaßnahmen bei manuellen Karteien und automatisierten Dateien (Zugangskontrolle, Zugriffsschutzprogramm, Zugriffsberechtigung, Aufbewahrungspflichten).

In Anlage 6 wird die Datenübermittlung innerhalb und außerhalb des öffentlichen Bereichs geregelt. Datenübermittlung bedeutet dabei das Bekannt geben gespeicherter Daten an Dritte durch Übermittlung über eine Datenleitung oder auf einem Datenträger und mündliche. oder schriftliche Auskünfte. Die Schule muss bei der Datenübermittlung innerhalb des öffentlichen Bereichs in jedem Einzelfall die Zulässigkeit überprüfen. Die Schule ist berechtigt, öffentlichen Stellen Schulbesuchsbescheinigungen auszustellen und die Adressen der Schüler und Elternvertreter an schulübergreifende Elternvertreter weiterzugeben. Eine Datenübermittlung an Stellen oder Personen außerhalb des öffentlichen Bereichs ist grundsätzlich nur mit Zustimmung des Betroffenen zulässig, dies gilt z. B. auch für die Übermittlung von Adressdaten von Mitschülern und Lehrern an ehemalige Schüler.

6.3 Datenschutz mit ACCESS

Werden personenbezogene Daten in der Schule verarbeitet, dann ist diese zu Maßnahmen des Datenschutzes und der Datensicherheit verpflichtet. Dies gilt natürlich auch für eine Bücherei, in der die Verwaltung der Daten mit *Access* vorgenommen wird.

Das Datenbanksystem stellt hierzu einige Hilfen zur Verfügung, mit denen zumindest die Zugriffskontrolle erleichtert wird.

Ein einfacher Schutz der gesamten Datenbank wird dadurch gewährleistet, dass ein Passwort vereinbart werden kann. Hierzu ist im Menü *Extras* unter *Zugriffsrechte* der Befehl *Datenbankkennwort* anzuwählen. Es öffnet sich ein Fenster zur Eingabe des Passwortes, das dann in der Datenbank selbst gespeichert wird und bei jedem Öffnen einzugeben ist. Anschließend hat der Benutzer alle Rechte.

Den besseren Schutz bietet die Möglichkeit, verschiedene Benutzergruppen zu erstellen, denen bestimmte Rechte eingeräumt werden. Der Systembetreuer, Administrator genannt, sollte sämtliche Rechte erhalten. Die übrigen Nutzer können nun für die verschiedenen Objekte der Datenbank (Tabellen, Abfragen, Formulare usw.) Rechte erhalten, nachdem sie eingerichtet wurden. Bei den Rechten wird unterschieden nach

- Öffnen/Ausführen
- Entwurf lesen
- Entwurf ändern
- Verwalten
- Daten lesen
- Daten aktualisieren
- Daten einfügen
- Daten löschen.

Im einfachsten Fall wird man die beiden bereits vorgesehenen Gruppen *Administrator* und *Benutzer* verwenden. Die Bibliothekarin erhält im Wesentlichen alle Rechte, außer den Löschrechten an den Stammdaten. Durch Eingabe des Passwortes für den Administrator ist dann die Datenbank gegen unbefugten Zugriff und versehentliche Löschung von Daten gesichert. Weitergehende Datenschutzmaßnahmen sind durch Programmierung mit der eingebauten Programmiersprache VBA (Visual Basic for Applications) möglich: So könnte z.B. bei sensibleren Daten, als sie in der Bücherei vorkommen, die Rückkehr in das Hauptmenü bei „unbeaufsichtigtem Rechner“ zusätzlich abgesichert werden.

Aufgaben

Aufgabe 1:

Das BVG hat als Risiken der automatisierten Datenverarbeitung insbesondere genannt:

1. das Risiko der unbegrenzten Speicherung von Daten
2. das Risiko der jederzeitigen Abrufbarkeit von Daten, ohne Rücksicht auf Entfernungen
3. das Risiko der Verknüpfung verschiedener Datensammlungen zu einem Persönlichkeitsbild des einzelnen
4. das Risiko der unzureichenden Kontrollierbarkeit der Datenverarbeitungsprozesse durch den Betroffen.

a) Geben Sie für jedes der genannten Risiken ein Beispiel an!

b) Untersuchen Sie die angegebenen Gesetzestexte darauf hin, wie diese die genannten Risiken zu unterbinden versuchen!

Aufgabe 2:

Die zu Beginn des Kapitels aufgeführten Beispiele beinhalten Tätigkeiten von Personen. Untersuchen Sie diese, inwieweit ihr Verhalten den Datenschutzbestimmungen entspricht!

Aufgabe 3:

Unter dem Begriff *Datenabgleich* versteht man das Herausfiltern von Datensätzen mit gemeinsamen Attributen aus verschiedenen Dateien.

a) Über den Schülernamen könnte ein Abgleich zwischen schulischen Leistungen und Lesegewohnheiten in der Schülerbibliothek stattfinden. Ist ein solcher Abgleich nach den hessischen Regelungen zulässig? Spielt dabei die Art der Speicherung und Verarbeitung (Kartei, Datei) eine Rolle?

b) Eine Lehrerin verwaltet ihre Tutorendaten und die Noten ihrer Schüler auf dem heimischen Computer. Sie gleicht ihre Noten mit denen der Kollegen ab, um ihre Schüler besser beraten zu können. Ist diese Vorgehensweise nach den hessischen Regelungen zulässig?

Aufgabe 4:

Informieren Sie sich beim Datenschutzbeauftragten Ihrer Schule über Art und umfang der angewandten automatisierten Verfahren und seine/ihre Tätigkeit! Wie werden in Ihrer Schule die Datensicherungs-Vorschriften nach §10HDSG umgesetzt?

Aufgabe 5:

Für bestimmte Daten im Schulbereich gibt es vorgeschriebene Löschfristen. Beantragen Sie nach Einsicht in Ihre Schulakte nach §72 HSG und überprüfen Sie, inwieweit diese Löschung vorgenommen wurde! Welche Rechte besitzen Sie, falls die Löschung nicht vollzogen wurde? An wen können Sie sich wenden?

Aufgabe 6:

Ein häufiger Diskussionspunkt ist die Frage, ob die Eltern volljähriger Schülerinnen und Schüler über die Leistungen oder das Fernbleiben vom Unterricht von der Schule informiert

werden dürfen. Klären Sie diese Frage mithilfe der abgedruckten Auszüge aus den hessischen Bestimmungen über die Datenübermittlung von Schuldaten!

Aufgabe 7:

Das in den Kapiteln 2 bis 5 angesprochene Beispiel einer Bibliotheksverwaltung ist auf datenschutzrechtliche Aspekte zu untersuchen.

a) Wie lassen sich Ausleihvorgänge so verwalten, dass Jahresstatistiken erstellt werden können?

b) Welche Maßnahmen sind zu ergreifen, wenn das Bibliotheksverwaltungsprogramm samt seinen Daten in das Schulnetz integriert wird, über das ein Zugriff auf Schülerindividualdaten möglich ist?

Aufgabe 8:

Eine Schule möchte die gesamte Verwaltung modernisieren und hierzu ein schuleigenes Netzwerk einrichten, über das alle in der Schule anfallenden Daten verarbeitet werden. Welche Bestimmungen sind hierbei einzuhalten?

Aufgabe 9:

Die abgedruckten Gesetzestexte enthalten keine Paragraphen zum Datenschutzbeauftragten des Landes bzw. zu den internen Datenschutzbeauftragten der Schule. Informieren Sie sich über Rechte und Aufgaben dieser Personen!

Lösungsheft

Kapitel 6
–
Rechtliche Aspekte

Aufgaben

Aufgabe 1:

Das BVG hat als Risiken der automatisierten Datenverarbeitung insbesondere genannt:

1. das Risiko der unbegrenzten Speicherung von Daten
2. das Risiko der jederzeitigen Abrufbarkeit von Daten, ohne Rücksicht auf Entfernungen
3. das Risiko der Verknüpfung verschiedener Datensammlungen zu einem Persönlichkeitsbild des einzelnen
4. das Risiko der unzureichenden Kontrollierbarkeit der Datenverarbeitungsprozesse durch den Betroffen.

a) Geben Sie für jedes der genannten Risiken ein Beispiel an!

b) Untersuchen Sie die angegebenen Gesetzestexte darauf hin, wie diese die genannten Risiken zu unterbinden versuchen!

a) "Computer können nicht vergessen" ist ein geflügeltes Wort, das ein wichtiges Risiko der Datenverarbeitung beschreibt. Computer können aber auch sehr große Mengen an Daten speichern, so könnte ein lückenloses „Tagebuch" über sämtliche Jugendsünden, Lebensgewohnheiten, Vorlieben und Abneigungen usw. beliebig lange gespeichert werden. In der Schule können Ordnungsmaßnahmen und Fernbleiben vom Unterricht über viele Schuljahre und Schulwechsel hinweg gespeichert bleiben.
Über Datennetze lassen sich Informationen jederzeit abrufen, unabhängig vom Speicherort (z.B. Internet). Hiervon sind auch sensible Daten betroffen, z.B. Konto- oder Kreditkartennummern.
Würden die Daten einer Person aus allen Dateien zusammengeführt, erhielte man ein *Persönlichkeitsprofil*, das je nach Auswahl bei der Speicherung die Person charakterisiert und gleichzeitig verfälscht.
Sämtliche Vorgänge bei der automatisierten Datenverarbeitung sind - bis auf die Erhebung beim Betroffenen, und auch das geschieht oft unbemerkt im Zusammenhang mit anderen Tätigkeiten - nicht direkt beobachtbar und damit nur schwer kontrollierbar.

b)

c) Das BVG hat u.a. im Zusammenhang mit dem Volkszählungsurteil 1983 auch die Einrichtung eines bundeseinheitlichen Melderegisters einschließlich der Einführung einer Personenkennziffer untersagt, um die Erstellung von Persönlichkeitsprofilen nicht Vorschub zu leisten. Die Datenschutzgesetze enthalten Vorschriften über die Sperrung und Löschen von Daten, sofern sie nicht mehr benötigt werden (s. §19HDSG) und beugen dem „Datenhunger der Behörden" dadurch vor, daß die Erforderlichkeit (s. §11 HDSG) und eine strenge Zweckbindung (s. §13 HDSG) gefordert wird. Die Betroffenen können insofern eine Kontrolle ausüben, als sie ein Auskunftrecht besitzen und von der Speicherung benachrichtigt werden müssen (s. §18 HDSG). Für den Schulbereich ist die Art der Daten, die in der Schule verarbeitet werden dürfen, direkt benannt.

Aufgabe 2:

Die zu Beginn des Kapitels aufgeführten Beispiele beinhalten Tätigkeiten von Personen. Untersuchen Sie diese, inwieweit ihr Verhalten den Datenschutzbestimmungen entspricht!

Beispiel 1:
Die Weitergabe von personenbezogenen Daten an Dritte außerhalb des öffentlichen Bereichs, so auch an Ehemalige, ist im Schulbereich in Hessen unzulässig, sofern der Betroffene nicht zustimmt. Dies ist im Falle der Wiedersehensparty nicht gegeben, die Sekretärin handelt richtig. Die Weitergabe von Daten an die Polizei liegt innerhalb des öffentlichen Bereichs und ist zulässig, wenn dies zur Erfüllung ihrer Aufgaben erforderlich ist und die Datenübermittlung im Rahmen der ursprünglichen Zweckbestimmung erfolgt. Dies ist im Beispiel nicht gegeben. Das HDSG liefert aber in §12 die Ausnahmefälle, z.B. bei Abwehr erheblicher Nachteile für das Allgemeinwohl oder wenn Anhaltspunkte für eine Straftat vorliegen. Die Schule muß aber die Zulässigkeit sehr genau prüfen, Auskünfte darf nur der Schulleiter oder ein von ihm Beauftragter erteilen. Inwieweit die Sekretärin also zulässig handelt, hängt vom konkreten Einzelfall ab.

Beispiel 2:
Der Personalrat hat seine Forderungen im Einklang mit den Datenschutzbestimmungen vorgetragen, in der Schule wurden bisher nicht die Forderungen zur Datensicherheit (s. §10 HDSG) umgesetzt. In Anlage 5 der hessischen Verordnung von 1993 sind diese detailliert aufgeführt: Zugangskontrolle zum Computerraum, Installation von Sicherheitssoftware, Passwortvergabe und Überprüfung der Zugriffsberechtigung, Speicherung nur auf Disketten oder verschlüsselt auf Festplatte, Einrichtung spezieller Sichten auf die Datenbestände je nach Zugriffsberechtigung für verschiedene Nutzer.

Beispiel 3:
Im Beispiel geht es nicht um personenbezogene Daten, also nicht um Datenschutz. Die Schule ist aber nach dem Urheberrecht verpflichtet, alle technischen und organisatorischen Maßnahmen zu ergreifen, dass Raubkopieren unterbleibt. Dies lässt sich unter einem Standardbetriebssystem wie MS-DOS praktisch nicht erreichen, ohne die Funktionsfähigkeit zu beeinträchtigen. Ein Netzwerkbetriebssystem ist eine gute Lösung.

Beispiel 4:
Der Hinweis der Datenschutzbeauftragten ist richtig und entspricht den Vorschriften. Die Schule ist nicht berechtigt, außer den in der Anlage 1 zur hessischen Verordnung von 1993 genannten Daten weitere Schülerdaten zu verarbeiten, z.B. Lesegewohnheiten o.ä. Nach der Rückgabe von Büchern sind die Ausleihdaten, soweit sie personenbezogen sind, zu löschen (s. §7 der Verordnung). Dies schließt nicht aus, dass zu statistischen Zwecken anonymisierte Daten gespeichert bleiben, aus denen Förderverein oder Schulleitung Rückschlüsse über die Lesegewohnheiten und die Akzeptanz des Buchbestandes entnehmen können.
Eine Kontrolle der Arbeit der Bibliothekarin durch den Computer, auf dem die Bibliothek verwaltet wird, erscheint unzulässig. Da die Bibliothekarin im Auftrag einer öffentlichen Stelle handelt, gilt für sie das Datenschutzgesetz. Nach §12 HDSG ist die Zweckbindung der erhobenen Daten vorgeschrieben, Ausleih- oder Kontrolldaten zur Datensicherung dürfen nicht zu anderen Zwecken, hier zur Arbeitskontrolle, eingesetzt werden.

Aufgabe 3:

Unter dem Begriff Datenabgleich versteht man das Herausfiltern von Datensätzen mit gemeinsamen Attributen aus verschiedenen Dateien.

Über den Schülernamen könnte ein Abgleich zwischen schulischen Leistungen und Lesegewohnheiten in der Schülerbibliothek stattfinden. Ist ein solcher Abgleich nach den hessischen Regelungen zulässig? Spielt dabei die Art der Speicherung und Verarbeitung (Kartei, Datei) eine Rolle?

Eine Lehrerin verwaltet ihre Tutorendaten und die Noten ihrer Schüler auf dem heimischen Computer. Sie gleicht ihre Noten mit denen der Kollegen ab, um ihre Schüler besser beraten zu können. Ist diese Vorgehensweise nach den hessischen Regelungen zulässig?

a) Da bereits das Speichern von Lesegewohnheiten nach den hessischen Bestimmungen in der Schule nicht zulässig ist, wäre auch ein Datenabgleich verboten. Es spielt dabei keine Rolle, ob die Daten auf dem Computer oder in einer Kartei gespeichert sind - hier sind die hessischen Regelungen strenger als die des Bundesdatenschutzgesetzes.

b) Nach §2 der hessischen Verordnung von 1993 kann der Schulleiter die Verarbeitung von Schülerdaten auf dem heimischen Computer der Tutorin dieser auf Antrag genehmigen, wenn sie die entsprechenden Datensicherheitsmaßnahmen ergreift und sich schriftlich der Kontrolle des Hessischen Datenschutzbeauftragten unterwirft. Die Daten sind beschränkt auf solche, welche in den pädagogischen Verantwortungsbereich der Lehrerin fallen, hierzu können gehören: Name, Vorname, Jahrgangsstufe, Kursbezeichnung, Unterrichtsfächer, Ergebnisse schriftlicher Arbeiten. In der Aufzählung fehlen konkret die Semesternoten; eine Verarbeitung dieser ist also auch für den Tutor unzulässig! Die Daten müssen spätestens nach Beendigung des Schuljahres gelöscht werden.

Aufgabe 4:

Informieren Sie sich beim Datenschutzbeauftragten Ihrer Schule über Art und umfang der angewandten automatisierten Verfahren und seine/ihre Tätigkeit! Wie werden in Ihrer Schule die Datensicherungs-Vorschriften nach §10HDSG umgesetzt?

do so…

Aufgabe 5:

Für bestimmte Daten im Schulbereich gibt es vorgeschriebene Löschfristen. Beantragen Sie nach Einsicht in Ihre Schulakte nach §72 HSG und überprüfen Sie, inwieweit diese Löschung vorgenommen wurde! Welche Rechte besitzen Sie, falls die Löschung nicht vollzogen wurde? An wen können Sie sich wenden?

Vermerke über Ordnungsmaßnahmen müssen spätestens nach 2 Schuljahren gelöscht werden. Sollten sich trotzdem solche Daten in den Akten befinden, ist der Schulleiter/die Schulleiterin der Ansprechpartner, aber auch der schulinterne Datenschutzbeauftragte kann behilflich sein.

Aufgabe 6:

Ein häufiger Diskussionspunkt ist die Frage, ob die Eltern volljähriger Schülerinnen und Schüler über die Leistungen oder das Fernbleiben vom Unterricht von der Schule informiert werden dürfen. Klären Sie diese Frage mithilfe der abgedruckten Auszüge aus den hessischen Bestimmungen über die Datenübermittlung von Schuldaten!

Auch hier ist die Rechtslage eindeutig: Die Datenübermittlung ist nur mit Einwilligung des Betroffenen, also des Schülers, möglich.

Aufgabe 7:

Das in den Kapiteln 2 bis 5 angesprochene Beispiel einer Bibliotheksverwaltung ist auf datenschutzrechtliche Aspekte zu untersuchen.

Wie lassen sich Ausleihvorgänge so verwalten, dass Jahresstatistiken erstellt werden können?

Welche Maßnahmen sind zu ergreifen, wenn das Bibliotheksverwaltungsprogramm samt seinen Daten in das Schulnetz integriert wird, über das ein Zugriff auf Schülerindividualdaten möglich ist?

a) Nach Abschluss eines Ausleihvorgangs kann entweder der Name des Ausleihers gelöscht werden, um die übrigen Daten zu erhalten, oder diese werden in eine anonymisierte Datei übernommen, in der z.B. nur die Zahl der Ausleihen eines bestimmten Buches registriert werden.

b) Hier gelten die Bestimmungen der hessischen Verordnung von 1993, z.B. ist nach §4 die Vernetzung von Geräten zur Verarbeitung personenbezogener Daten mit Datenverarbeitungszwecken für Unterrichtszwecke untersagt. Im Übrigen gelten die Vorschriften zur Datensicherung und zum Datenschutz nach §10 HDSG bzw. die Erläuterungen hierzu in Anlage 5 der genannten Verordnung. Die Bibliothekarin darf nur über Passwort in das Netz gelangen und nur eine eigene Sicht auf personenbezogene Daten erhalten, die zur Erfüllung ihrer Aufgaben notwendig ist. Hierzu gehören sicherlich Name und Tutor bzw. Klassenlehrer der Ausleiher, eventuell Jahrgangsstufe. Ein schreibender Zugriff auf diese Daten ist nicht zulässig.

Aufgabe 8:

Eine Schule möchte die gesamte Verwaltung modernisieren und hierzu ein schuleigenes Netzwerk einrichten, über das alle in der Schule anfallenden Daten verarbeitet werden. Welche Bestimmungen sind hierbei einzuhalten?

Die Vorschriften gelten analog zu denen aus Aufgabe 7. Das Netzwerkbetriebssystem und das Datenbanksystem müssen für die einzelnen Benutzergruppen eindeutige Schreib- und Leserechte definieren, z.B. für Schulleitung, Studienleiter, Sekretariat usw. Entsprechende Software, die das ganze Spektrum der Schulverwaltung integriert abdeckt, existiert bisher nicht.

Aufgabe 9:

Die abgedruckten Gesetzestexte enthalten keine Paragraphen zum Datenschutzbeauftragten des Landes bzw. zu den internen Datenschutzbeauftragten der Schule. Informieren Sie sich über Rechte und Aufgaben dieser Personen!

Der Hessische Datenschutzbeauftragte hat sein Büro in 65189 Wiesbaden, Uhlandstr. 4. Hier sind Informationsmaterialien zu erhalten. Das HDSG regelt die Aufgaben und Rechte des Datenschutzbeauftragten in den Paragraphen 21 bis 31, die hier nicht abgedruckt sind.

Projekt: Umsetzung einer Bibliotheksverwaltung

Ziel: Entwicklung einer Bibliotheksverwaltung von der Planung bis zur Produktreife, inklusive Anwenderhandbuch (für die Bibliothekarin als Laien!) und Problemlösungsbericht (für den Lehrer als Experten).

Zusatz: Selbständiges Erarbeiten der nötigen Access-„Detailkenntnisse“

IST - ANALYSE

der Martin-Niemöller Schulbibliothek

Die Martin-Niemöller Schulbibliothek besitzt ungefähr 10.000 - 12.000 Bücher. Pro Tag werden bei der Bibliothekarin circa 30 Bücher ausgeliehen.

Damit die Schüler die Möglichkeit haben, nach bestimmten Buchtiteln und Autoren zu suchen, steht außerdem in der Bibliothek ein Stich- und Schlagwortkatalog bereit.

Allgemeine Informationen, wie die Ausweisnummer, der Schülervor- und Schülernachname werden in einer Kartei gesammelt. Für die Bücher wurde eine Bücherkarte angelegt, die Informationen über das Fachgebiet, den Titel, den Autor bzw. Herausgeber, den Verlag, den Verlagsort, der ID-Nummer und das Erscheinungsdatum des Buches enthält. Des weiteren vergibt die Bibliothekarin eine selbst gewählte Nummer im Format xxxxx.jj - der eigentlichen Buchnummer gefolgt von der aktuellen Jahreszahl.

Bei der Ausleihe eines Buches durch einen Schüler füllt die Bibliothekarin eine Ausweiskarte aus, auf der die Buchnummer und die Schülerausweisnummer eingetragen werden. Für das auszuleihende Buch wird zudem eine Fristkarte benötigt, auf der das Rückgabedatum steht. Die Fristkarte wird dem Schüler mitgeben, die Ausweiskarte behält die Bibliothekarin.

Wird ein Buch zurückgegeben, wird der Schüler von der Ausweiskarte gestrichen, diese wieder in das Buch zurück gesteckt und die Fristkarte entfernt.

ANFORDERUNGSANALYSE

Die Bibliotheks-Datenbank soll über ein Formular verfügen, in dem die Schülerinformationen aufgenommen werden können. Diese müssen aus dem Schülernachnamen, dem Vornamen und der Ausweisnummer bestehen. Außerdem muss sichergestellt werden, dass eine Ausweisnummer immer nur einmal zugeordnet wird.

Für Buchinformationen soll ein Formular vorhanden sein, das eine eindeutige Buchnummer, die nicht doppelt vorkommen darf, das Sachgebiet, die ID-Nummer, den Titel, den Autor bzw. Herausgeber, den Verlag, den Verlagsort und das Erscheinungsdatum zur Verfügung stellt.

Die Aufnahme der Bücher soll erst mit dem Ausleihen eines solchen erfolgen, da die Eingabe von 12.000 Bücher nicht zu bewerkstelligen ist.

Bei der Ausleihe sollen nur die Ausweisnummer und die Buchnummer angegeben werden. Zusatzinformationen, die den richtigen Schüler und das richtige Buch verifizieren, sollen ebenfalls vorhanden sein (Schülername, Buchtitel, etc.).

Bei der Buchrückgabe, soll nur die Buchnummer eingeben werden - existiert diese, sollen alle dazu erhältlichen Informationen angezeigt werden, diese sollen aber nicht veränderbar sein. Nach der Buchrückgabe sollen aus Datenschutzgründen die Schülerinformationen gelöscht werden; es soll aber eine spätere Auswertung in Form einer Statistik möglich sein.

Außerdem muss die Datenbank Anfragen beantworten können, damit Schüler die länger als drei Monate keine ausgeliehenen Bücher abgeben haben angezeigt und gemahnt werden können. Diese will die Bibliothekarin vierteljährlich ausführen. Des weiteren soll die Möglichkeit gegeben werden Anfragen über ein Fachgebiet, den Autor und den Buchtitel zu stellen. Eine Abfrage, die alle Schüler anzeigt, die noch Bücher ausgeliehen haben, soll auch vorhanden sein. Zusätzlich soll konkret nach Schülernamen gesucht werden können.

Die Datenbank soll benutzerfreundlich sein und dem SAA-Standard entsprechen. Eine schnelle Einarbeitung soll ebenfalls garantiert sein.

UMSETZUNG der IST-Analyse

Für die Umsetzung der IST-Analyse wurde folgendes Entity Relationshipmodell entwickelt.

Tipp: … es genügen für den Anfang 2 Entities und 1 Beziehungsrelation

ER-Modell

Makros in Access

(auf Ihre Tabellenstruktur anzupassen, hier also nur als Anregung zu verstehen!)

Bei der Umsetzung in Access wurden zum Abfangen von Fehlern und Doppeleingaben Makros eingeführt. Mit diesen ist es möglich die Datenbank komfortabler und leichter bedienbar zu machen. Im Folgenden soll die Wirkungsweise eines Makros erläutert werden, welches bei verlassen eines bestimmten Feldes aufgerufen wird

```
Sub Buchnummer_AfterUpdate()
' Diese Prozedur versucht die übereinstimmende Buchnummer zu
finden.
' Wird der Datensatz gefunden, wird eine Fehlermeldung ausgegeben
' Wird er nicht gefunden, wird der Datensatz akzeptiert.

   Dim Kriterien As String       ' Das Argument der Methode
"FindFirst".
   Dim DSG1 As Recordset         ' Datensatzgruppe für den
Suchvorgang.
   Dim SBuch As String           ' Die Buchnummer, nach der gesucht
wird.

   Set DSG1 = Me.RecordsetClone ' Formulieren der Kriterien.
   SBuch = Chr$(34) & Screen.ActiveControl & Chr$(34)
   Kriterien = "[Buchnummer] =" + SBuch ' Durchführen des
Suchvorgangs.
   DSG1.FindFirst Kriterien
   If DSG1.NoMatch = False Then
      fehler = MsgBox("Das eingegebene Buch wird bereits
ausgeliehen!", 16,
      "Schulbibliothek")
      Forms![Ausleihe: Hauptformular]![Buchnummer] = "DOPPELT!"
      DoCmd.GoToControl "Ausleihdatum"
      DoCmd.GoToControl "Buchnummer"
   End If
   DSG1.Close
End Sub
```

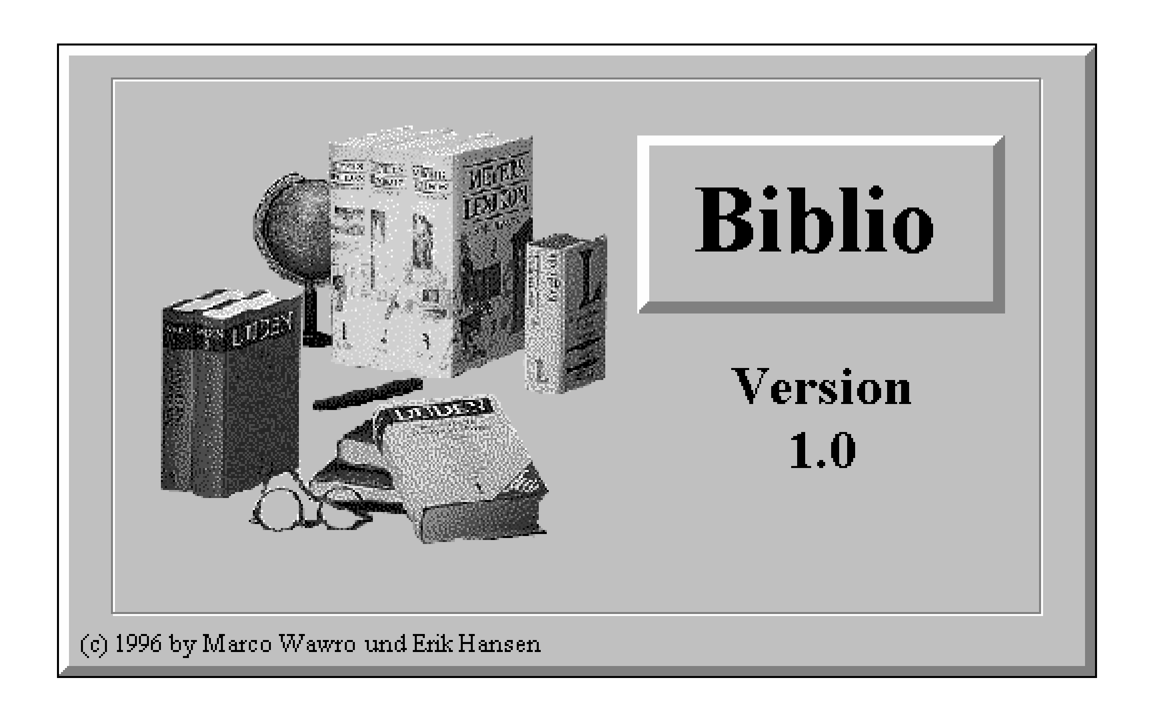
Obiges Makro überprüft nach Eingabe einer Buchnummer ob das dazugehörige Buch schon ausgeliehen ist und gibt falls schon verliehen, eine Fehlermeldung aus, und setzt das Feld Buchnummer auf DOPPELT!.

Makros verleihen einer Datenbank mehr Flexibilität, da diese eine kleine Programmiersprache in Access ist. Mit ihr können sogar DLL-Aufrufe realisiert werden; damit wird der Funktionsumfang sogar noch erheblich gesteigert.

Wie bindet man Makros nun in sein Formular oder seinen Bericht ein?

Nachdem die Entwurfansicht des entsprechenden Formulars/Berichts gewählt wurde, kann man einen Button, ein Feld oder jedes andere Objekt, welches im Eigenschaftsmenü Ereignisse zulässt, mit einem Makro belegen. Dazu klickt man z. B. bei einem Button das Ereignis *Beim Klicken* an und wählt *[Ereignisprozedur]* aus. Nun kann man mit dem eigentlichen Herstellen des Makros beginnen, indem man den auszuführenden Code eingibt. Nach erfolgreicher Kompilierung, wird jedes Mal, wenn der Button im Formular betätigt wird, die Ereignisprozedur *Beim Klicken* und der darin stehende Code ausgeführt (siehe Buttons in der Schulbibliothek).

Als Anregung können folgende Bildschirmfotos dienen, von denen Sie in Ihrem Handbuch auch reichlich Gebrauch machen müssen.:

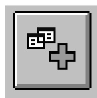


Handbuch zur Bibliotheksverwaltung der

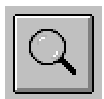
Martin-Niemöller-Schule



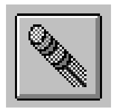
Datensatz-Navigation



Ein neuer Datensatz wird angelegt.



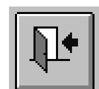
Ein Suchfenster erscheint.



Die zuvor gemachten Eingaben werden rückgängig gemacht.



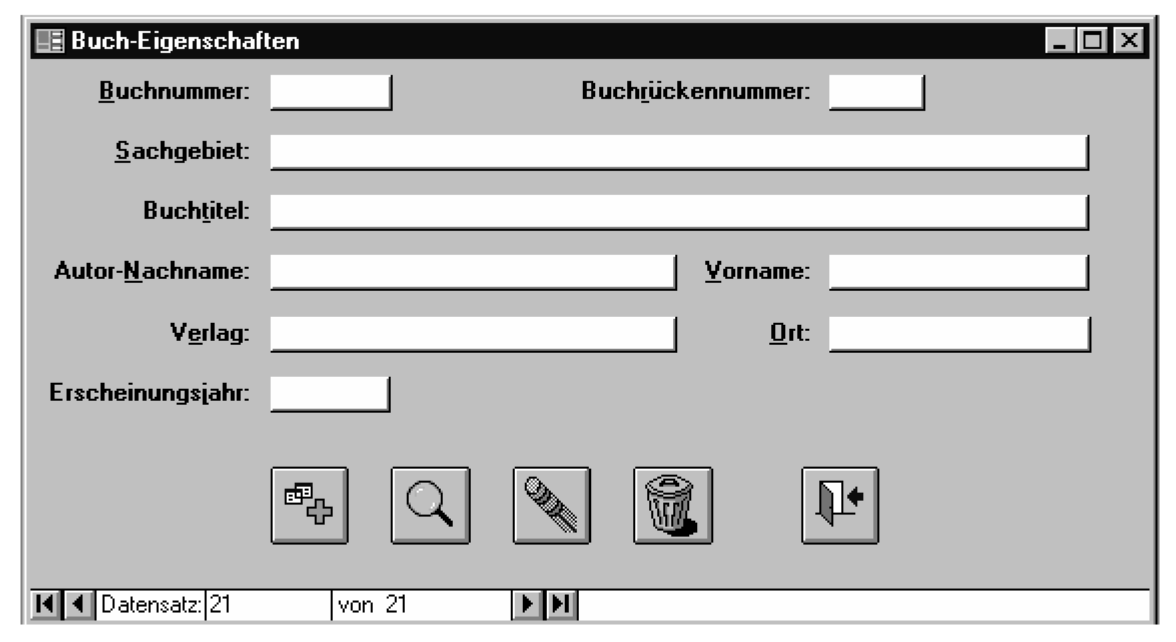
Der aktuelle Datensatz wird nach Rückfrage unwiderruflich vom Datenträger gelöscht.



Die Eingabemaske wird verlassen und der Navigator erscheint wieder. Zuvor getätigte Eingaben werden automatisch gespeichert.

Bucheigenschaften

In die Eingabemaske werden die auszuleihenden Bücher eingetragen. Nach Aufruf über den entsprechenden Button im Navigator wird in der Eingabemaske automatisch ein neuer Datensatz angelegt, um ein Buch einzutragen.



Bucheigenschaften

Schülereigenschaften

In die Eingabemaske werden die Schüler, die ein Buch ausleihen und noch nicht in der Datenbank stehen, eingetragen. Nach Aufruf über den entsprechenden Button im Navigator wird in der Eingabemaske automatisch ein neuer Datensatz angelegt, um ein Schüler einzutragen.



Schülereigenschaften

Ausleihe

In die Eingabemaske werden die Schüler, die ein Buch ausleihen möchten eingetragen. Nach Aufruf über den entsprechenden Button im Navigator wird in der Eingabemaske automatisch ein neuer Datensatz angelegt, um eine Ausleihe durchzuführen und einzutragen. Für das Ausleihdatum wird das aktuelle Tagesdatum eingesetzt. Blau unterlegte Felder dienen nur zur Informationsangabe - sie können nicht verändert werden.



Buchausleihe

Rückgabe

Wird ein Buch zurückgegeben, wird zuvor auf den entsprechenden Button im Navigator geklickt. Anschließend erscheint ein kleines Fenster, das nach der Buchnummer fragt. Wird die richtige Buchnummer eingeben, erscheinen, wie die untere Abbildung zeigt, Rückgabeinformationen.

In diesem Rückgabeformular können keine Felder geändert werden. Außerdem sind nur zwei grafische Buttons, auch Icons genannt, auf dem Bildschirm vorhanden. Zum einen das Tür-Icon wie bei den anderen Formularen, zum anderen folgendes Rückgabe-Icon

Nach anklicken dieses Icons wird ein ausgeliehenes Buch in der Tabelle Ausleihe als zurückgegeben markiert und es wird zum Navigator zurückgesprungen. Des weiteren wird die Ausweisnummer des Ausleihers auf den Wert Null (0) gesetzt, da es wegen Datenschutzgründen nicht zulässig ist, einen Schüler auf Dauer zu speichern. Eine spätere Statistik ist durch diesen Vorgang nach wie vor möglich, da der Rest der Informationen erhalten bleibt.



Buchrückgabe

Abfragen...

Um mehrere Informationen auf einen Blick zu erhalten gibt es Abfragen. Diese werden über die Schaltfläche Abfragen... im Navigator aufgerufen. Darauf hin erscheint ein Untermenü mit den Auswahlmöglichkeiten

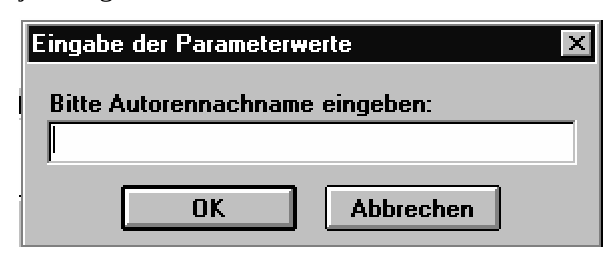
Sachgebietssuche	sucht nach einem Sachgebiet
Buchtitel suchen	sucht einen Buchtitel
Autorensuche	sucht einen Autor
Ausleiher suchen	sucht nach einem Schüler, der noch Bücher ausgeliehen hat
Ausleihe > 3 Monate	zeigt alle Schüler an, die ihr Buch mehr als 3 Monate besitzen
Buchnummer suchen	zeigt Informationen über eine Buchnummer an
Zurück zum Hauptmenü	springt zurück zum Navigator



Abfrage-Untermenü

In diesem Untermenü können spezifische Informationen zu den einzelnen Gebieten gesucht und angezeigt werden.

Wird eine bestimmte Schaltfläche angeklickt, erscheint eine Abfragebox, die Eingaben zum jeweiligen Gebiet erwartet.



Abfragebox

Projekt: Umsetzung einer Bibliotheksverwaltung

Ziel: Entwicklung einer Bibliotheksverwaltung von der Planung bis zur Produktreife, inklusive Anwenderhandbuch (für die Bibliothekarin als Laien!) und Problemlösungsbericht (für den Lehrer als Experten).

Zusatz: Selbständiges Erarbeiten der nötigen Access-„Detailkenntnisse“

IST - ANALYSE

der Martin-Niemöller Schulbibliothek

Die Martin-Niemöller Schulbibliothek besitzt ungefähr 10.000 - 12.000 Bücher. Pro Tag werden bei der Bibliothekarin circa 30 Bücher ausgeliehen.

Damit die Schüler die Möglichkeit haben, nach bestimmten Buchtiteln und Autoren zu suchen, steht außerdem in der Bibliothek ein Stich- und Schlagwortkatalog bereit.

Allgemeine Informationen, wie die Ausweisnummer, der Schülervor- und Schülernachname werden in einer Kartei gesammelt. Für die Bücher wurde eine Bücherkarte angelegt, die Informationen über das Fachgebiet, den Titel, den Autor bzw. Herausgeber, den Verlag, den Verlagsort, der ID-Nummer und das Erscheinungsdatum des Buches enthält. Des weiteren vergibt die Bibliothekarin eine selbst gewählte Nummer im Format xxxxx.jj - der eigentlichen Buchnummer gefolgt von der aktuellen Jahreszahl.

Bei der Ausleihe eines Buches durch einen Schüler füllt die Bibliothekarin eine Ausweiskarte aus, auf der die Buchnummer und die Schülerausweisnummer eingetragen werden. Für das auszuleihende Buch wird zudem eine Fristkarte benötigt, auf der das Rückgabedatum steht. Die Fristkarte wird dem Schüler mitgeben, die Ausweiskarte behält die Bibliothekarin.

Wird ein Buch zurückgegeben, wird der Schüler von der Ausweiskarte gestrichen, diese wieder in das Buch zurück gesteckt und die Fristkarte entfernt.

ANFORDERUNGSANALYSE

Die Bibliotheks-Datenbank soll über ein Formular verfügen, in dem die Schülerinformationen aufgenommen werden können. Diese müssen aus dem Schülernachnamen, dem Vornamen und der Ausweisnummer bestehen. Außerdem muss sichergestellt werden, dass eine Ausweisnummer immer nur einmal zugeordnet wird.

Für Buchinformationen soll ein Formular vorhanden sein, das eine eindeutige Buchnummer, die nicht doppelt vorkommen darf, das Sachgebiet, die ID-Nummer, den Titel, den Autor bzw. Herausgeber, den Verlag, den Verlagsort und das Erscheinungsdatum zur Verfügung stellt.

Die Aufnahme der Bücher soll erst mit dem Ausleihen eines solchen erfolgen, da die Eingabe von 12.000 Bücher nicht zu bewerkstelligen ist.

Bei der Ausleihe sollen nur die Ausweisnummer und die Buchnummer angegeben werden. Zusatzinformationen, die den richtigen Schüler und das richtige Buch verifizieren, sollen ebenfalls vorhanden sein (Schülername, Buchtitel, etc.).

Bei der Buchrückgabe, soll nur die Buchnummer eingeben werden - existiert diese, sollen alle dazu erhältlichen Informationen angezeigt werden, diese sollen aber nicht veränderbar sein. Nach der Buchrückgabe sollen aus Datenschutzgründen die Schülerinformationen gelöscht werden; es soll aber eine spätere Auswertung in Form einer Statistik möglich sein.

Außerdem muss die Datenbank Anfragen beantworten können, damit Schüler die länger als drei Monate keine ausgeliehenen Bücher abgeben haben angezeigt und gemahnt werden können. Diese will die Bibliothekarin vierteljährlich ausführen. Des weiteren soll die Möglichkeit gegeben werden Anfragen über ein Fachgebiet, den Autor und den Buchtitel zu stellen. Eine Abfrage, die alle Schüler anzeigt, die noch Bücher ausgeliehen haben, soll auch vorhanden sein. Zusätzlich soll konkret nach Schülernamen gesucht werden können.

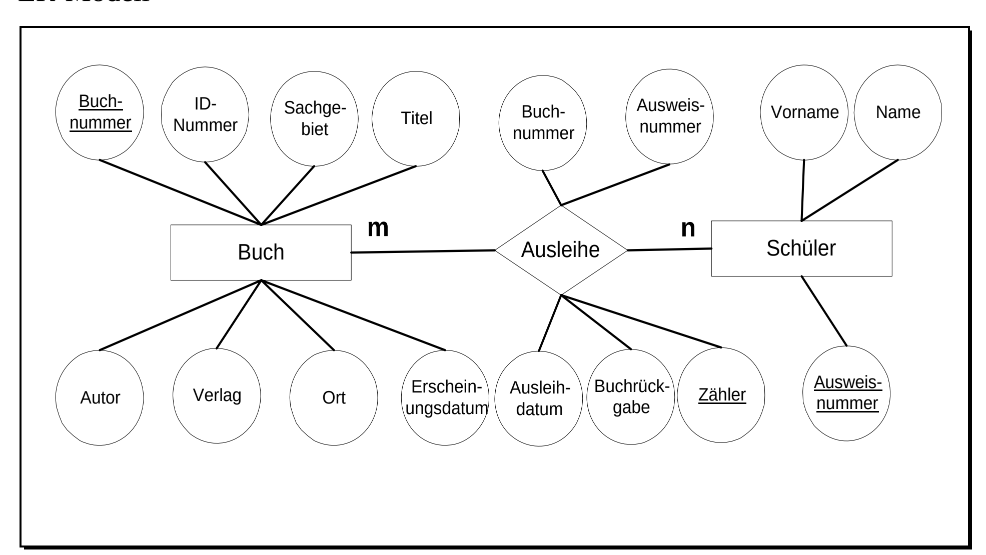
Die Datenbank soll benutzerfreundlich sein und dem SAA-Standard entsprechen. Eine schnelle Einarbeitung soll ebenfalls garantiert sein.

UMSETZUNG der IST-Analyse

Für die Umsetzung der IST-Analyse wurde folgendes Entity Relationshipmodell entwickelt.

Tipp: … es genügen für den Anfang 2 Entities und 1 Beziehungsrelation

ER-Modell



Zur Umsetzung in eine Datenbank werden drei Tabellen benötigt. Unterstrichene Attribute sind Primärschlüssel, sie dürfen nicht doppelt vorkommen und keine Nullwerte beinhalten.

Tabelle Buch

Buch-nummer	ID-Nummer	Sach-gebiet	Titel	Autor	Verlag	Ort	Erscheinungs jahr
161-95	Gcr	Frauen	Geschichte der Frau	Duby	Campus	Frankfurt	1993-1995
175-95	Vg	Drogen	Hasch	Mann	Fischer	Frankfurt	1994
238-95	Gkl 1	EU	Handlexikon der EU	Mickel	Omnia	Köln	1994

Tabelle Schüler

Ausweisnummer	Schülername	Schülervorname
0000	Anonymität	wegen Datenschutz
0001	Maffiose	Peter
0002	Nichtnutz	Jürgen
0003	Laber	Andreas

Um eine spätere Auswertung der ausgeliehenen Bücher zu vollziehen, muss in der Tabelle Schüler ein Datensatz vorhanden sein, der allgemein gehalten wird, wie z.B. 0000-Anonymität-wegen Datenschutz.

Tabelle Ausleihe

Zähler	Ausweisnummer	Buchnummer	Ausleihdatum	Buch zurückgegeben
1	0000	165-95	01.01.1996	Ja
2	0001	175-95	25.06.1996	Nein
3	0003	165-95	10.11.1996	Nein

Damit eine Statistik bei der Ausleihe möglich werden kann, wird ein Zähler als Primärschlüssel gesetzt. So werden doppelte Eingaben bei der Ausleihe verhindert.

Um die im Entity Relationshipmodell entwickelte *n* zu *m* Beziehung erstellen zu können, muss bei der Umsetzung in MS-Access diese in zwei *1* zu *n* Beziehungen aufgespalten werden. Dazu muss eine *1* zu *n* Beziehung zwischen Buch - Ausleihe und eine selbige zwischen Schüler - Ausleihe gezogen werden.

Makros in Access

Bei der Umsetzung in Access wurden zum Abfangen von Fehlern und Doppeleingaben Makros eingeführt. Mit diesen ist es möglich die Datenbank komfortabler und leichter bedienbar zu machen. Im Folgenden soll die Wirkungsweise eines Makros erläutert werden, welches bei verlassen eines bestimmten Feldes aufgerufen wird

```
Sub Buchnummer_AfterUpdate()
' Diese Prozedur versucht die übereinstimmende Buchnummer zu
finden.
' Wird der Datensatz gefunden, wird eine Fehlermeldung ausgegeben
' Wird er nicht gefunden, wird der Datensatz akzeptiert.

   Dim Kriterien As String      ' Das Argument der Methode
"FindFirst".
   Dim DSG1 As Recordset        ' Datensatzgruppe für den
Suchvorgang.
   Dim SBuch As String          ' Die Buchnummer, nach der gesucht
wird.

   Set DSG1 = Me.RecordsetClone ' Formulieren der Kriterien.
   SBuch = Chr$(34) & Screen.ActiveControl & Chr$(34)
   Kriterien = "[Buchnummer] =" + SBuch ' Durchführen des
Suchvorgangs.
   DSG1.FindFirst Kriterien
   If DSG1.NoMatch = False Then
      fehler = MsgBox("Das eingegebene Buch wird bereits
ausgeliehen!", 16,
      "Schulbibliothek")
      Forms![Ausleihe: Hauptformular]![Buchnummer] = "DOPPELT!"
      DoCmd.GoToControl "Ausleihdatum"
      DoCmd.GoToControl "Buchnummer"
   End If
   DSG1.Close
End Sub
```

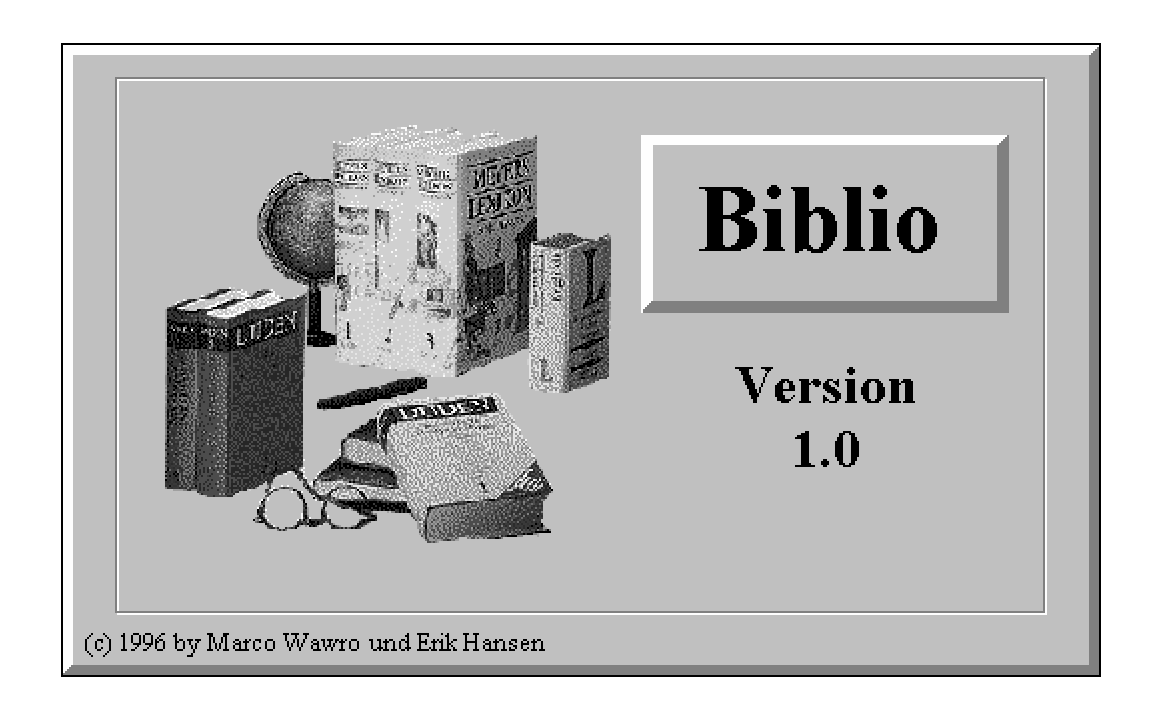
Obiges Makro überprüft nach Eingabe einer Buchnummer ob das dazugehörige Buch schon ausgeliehen ist und gibt falls schon verliehen, eine Fehlermeldung aus, und setzt das Feld Buchnummer auf DOPPELT!.

Makros verleihen einer Datenbank mehr Flexibilität, da diese eine kleine Programmiersprache in Access ist. Mit ihr können sogar DLL-Aufrufe realisiert werden; damit wird der Funktionsumfang sogar noch erheblich gesteigert.

Wie bindet man Makros nun in sein Formular oder seinen Bericht ein?

Nachdem die Entwurfansicht des entsprechenden Formulars/Berichts gewählt wurde, kann man einen Button, ein Feld oder jedes andere Objekt, welches im Eigenschaftsmenü Ereignisse zulässt, mit einem Makro belegen. Dazu klickt man z. B. bei einem Button das Ereignis *Beim Klicken* an und wählt *[Ereignisprozedur]* aus. Nun kann man mit dem eigentlichen Herstellen des Makros beginnen, indem man den auszuführenden Code eingibt. Nach erfolgreicher Kompilierung, wird jedes Mal, wenn der Button im Formular betätigt wird, die Ereignisprozedur *Beim Klicken* und der darin stehende Code ausgeführt (siehe Buttons in der Schulbibliothek).

Marco Wawro, Eric Hansen (LK Informatik Jg. 12 1996)



Handbuch zur Bibliotheksverwaltung der

Martin-Niemöller-Schule

Inhaltsverzeichnis



Einführung



Start der Bibliotheksverwaltung

Die Bibliotheksverwaltung wird im Programmanager über das Symbol *Bibliothek-Verwaltung* in der Gruppe *Bibliothek* mit einem Doppelklick gestartet. Anschließend startet Access und der Bibliotheksnavigator wird geladen.

Allgemeine Hinweise zur Oberfläche und den Bedienelementen

Es wurde in der Bibliotheksverwaltung so weit wie möglich versucht einheitliche Eingabemasken zu gestalten, so daß sich der Benutzer nicht ständig auf eine andere Benutzerführung einstellen muß. Die Bedienung kann sowohl mit der Maus als auch mit der Tastatur erfolgen. Für die Tastaturbedienung stehen sogenannte Shortcuts, zu deutsch Abkürzungstasten, in jeder Eingabemaske zur Verfügung. Ein Shortcut wird ausgeführt, indem die ALT-Taste und der unterstrichene Buchstabe gedrückt wird.

In der Bibliotheksverwaltungen werden zusätzliche Informationen in der Statuszeile angezeigt:

Die Eingabemaske für die Bucheingabe wird gestartet.

Abbildung 1: Statuszeile

Da es, wie bereits weiter oben erwähnt, einheitliche Eingabemasken gibt, werden die Elemente erklärt die in jeder Eingabemaske identisch sind.

Am unteren Rand jeder Eingabemaske befinden sich Navigations-Recorder-Tasten, die von links nach rechts gelesen folgende Funktion erfüllen:

- Gehe zu erstem Datensatz
- Gehe einen Datensatz zurück
- Gehe einen Datensatz vorwärts
- Gehe zum letzten Datensatz

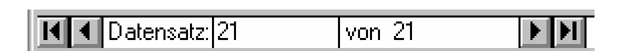
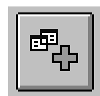


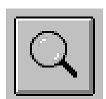
Abbildung 2: Datensatz-Navigation

Zwischen den Recorder-Tasten befinden sich allgemeine Angaben zum Datensatz wie Gesamtanzahl, o.a.

Verschiedene Buttons erleichtern das Arbeiten mit den Eingabemasken. Durch einen Mausklick werden die dazugehörigen Aktionen ausgeführt:



Ein neuer Datensatz wird angelegt.



Ein Suchfenster erscheint.

Das Feld “Suchen nach” enthält den Datensatz, der zu suchen ist, z.B. “Burkert”. Hellblau unterlegte (gesperrte) Felder werden nicht durchsucht.

Im Feld “Vergleichen” kann zwischen 3 Optionen gewählt werden, nämlich

1. <u>Teil des Feldinhaltes</u>

   Sucht nach jedem beliebigen Auftreten der Textzeichenfolge. Wenn Sie z.B. nach "berg" suchen, finden Sie sowohl "Bergen" als auch "Heidelberg".

2. <u>Gesamter Feldinhalt</u>

   Erkennt nur dann eine Übereinstimmung, wenn die Textzeichenfolge dem gesamten Inhalt des Feldes entspricht. Wenn Sie z.B. nach "Köln" suchen, finden Sie nur "Köln", nicht jedoch "Kölner Str.".

3. <u>Anfang des Feldinhaltes</u>

   Sucht nach der Zeichenfolge am Anfang eines Feldes. Wenn Sie z.B. nach "Berg" suchen, finden Sie nur "Bergen", nicht jedoch "Heidelberg".

Das Feld “Suchbereich” ist abgeblendet, es ist also für den Benutzer nicht zugänglich. Es dient dazu auszuwählen, ob das aktuelle Feld oder alle Felder durchsucht werden sollen. Es ist standardmäßig auf “Alle Felder” geschaltet, was Sie auch am ausgefüllten Optionsfeld erkennen können.

Das Feld “Suchrichtung” entscheidet, ob nach dem Datensatz aufwärts oder abwärts gesucht werden soll.

Das Feld “Groß-/Kleinschreibung” entscheidet ob die Groß- und Kleinschreibung beachtet werden soll, z.B. daß der Suchbegriff Burkert nur gefunden werden kann, wenn er genauso wie die Suchmaske geschrieben wurde; das Wort “BuRkErT” würde bei eingeschalteter Option nicht gefunden werden, da die Schreibung des Wortes anders lautet, als die in der Suchmaske. Das Feld “Formatierungen beachten” ist abgeblendet und standardmäßig angeschaltet.

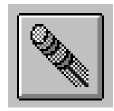
Die Daten müssen dem angezeigten Format des Suchtextes entsprechen.

Datumsangaben werden in der Datenbank als Zahlen gespeichert; Sie können sie aber z.B. im Format "05. Jan. 92" anzeigen. Um dann nach Datensätzen aus dem Monat Januar 1992 zu suchen, kann die Suchzeichenfolge in Form von "*Jan. 92" eingegeben werden.

Die Schaltflächen “Suchen”, “Weitersuchen” und “Schließen” sprechen für sich, sie bedürfen keiner weiteren Erklärung.

Sofern sich mehrere Datensätze in der Datenbank befinden, muß, um alle Datensätze angezeigt zu bekommen, die Schaltfläche “Weitersuchen” betätigt werden.

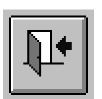
Alternativ dazu existiert eine Abfrage, die die gewünschten Datensätze beinhaltet. Diese befinden sich im Unterpunkt "Abfragen...". Doch dazu später mehr.



Die zuvor gemachten Eingaben werden rückgängig gemacht.



Der aktuelle Datensatz wird nach Rückfrage unwiderruflich vom Datenträger gelöscht.



Die Eingabemaske wird verlassen und der Navigator erscheint wieder. Zuvor getätigte Eingaben werden automatisch gespeichert.

Bedienung der Bibliotheksverwaltung

1

Navigationszentrum

Das Navigationszentrum ist wie der Name schon besagt die Steuerzentrale der Bibliotheksverwaltung, die aus den Buttons

- Bucheigenschaften
- Schülereigenschaften
- Ausleihe
- Rückgabe
- Abfragen...
- Beenden

besteht. Nach anklicken eines Menüpunktes wird in das entsprechende Formular bzw. in das entsprechende Abfrageuntermenü verzweigt.

Bucheigenschaften

In die Eingabemaske werden die auszuleihenden Bücher eingetragen. Nach Aufruf über den entsprechenden Button im Navigator wird in der Eingabemaske automatisch ein neuer Datensatz angelegt, um ein Buch einzutragen.

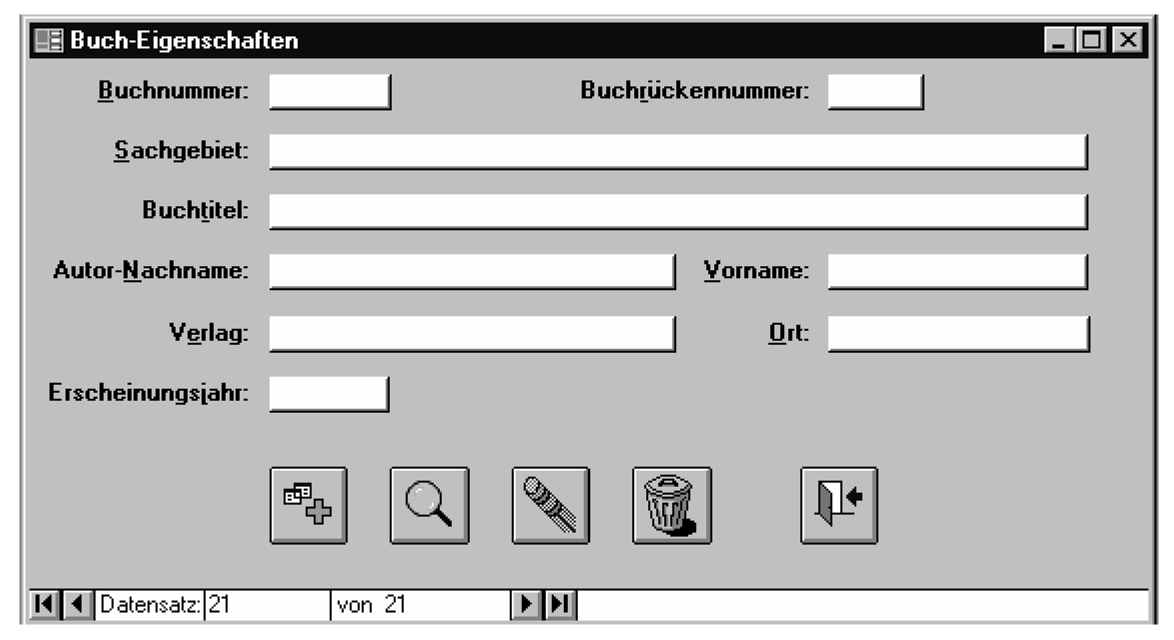


Abbildung 3: Bucheigenschaften

Schülereigenschaften

In die Eingabemaske werden die Schüler, die ein Buch ausleihen und noch nicht in der Datenbank stehen, eingetragen. Nach Aufruf über den entsprechenden Button im Navigator wird in der Eingabemaske automatisch ein neuer Datensatz angelegt, um ein Schüler einzutragen.



Abbildung 4: Schülereigenschaften

Ausleihe

In die Eingabemaske werden die Schüler, die ein Buch ausleihen möchten eingetragen. Nach Aufruf über den entsprechenden Button im Navigator wird in der Eingabemaske automatisch ein neuer Datensatz angelegt, um eine Ausleihe durchzuführen und einzutragen. Für das Ausleihdatum wird das aktuelle Tagesdatum eingesetzt. Blau unterlegte Felder dienen nur zur Informationsangabe - sie können nicht verändert werden.

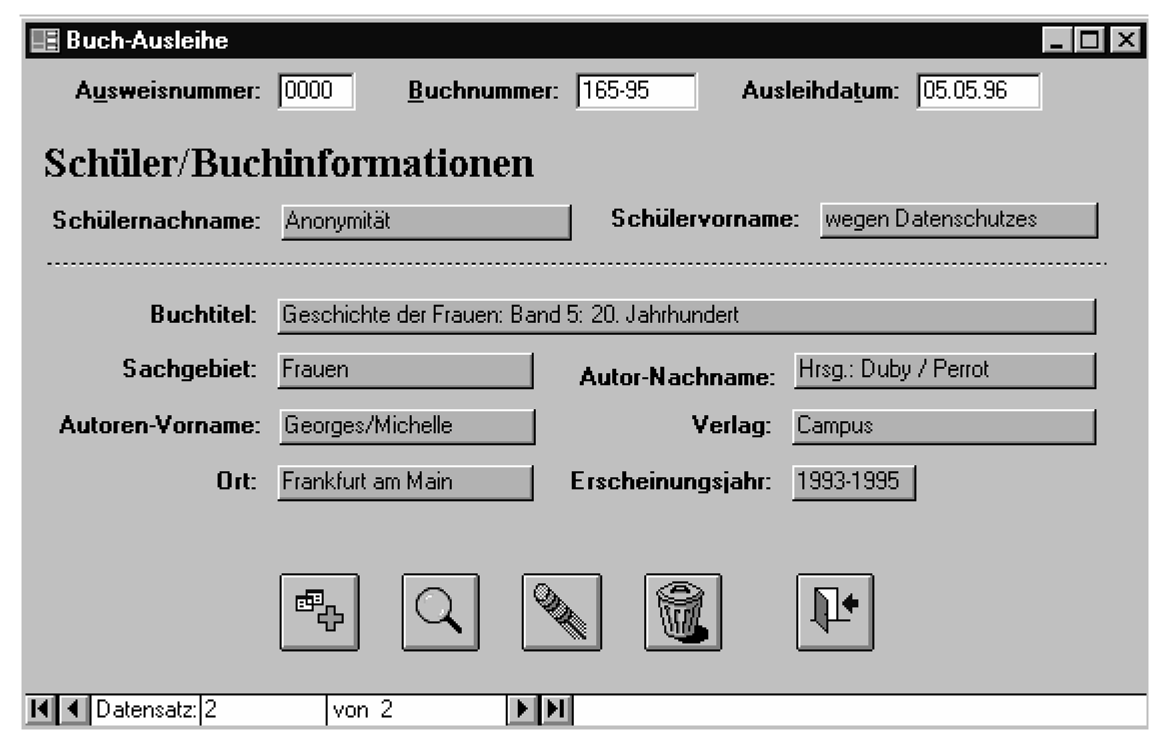
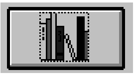


Abbildung 5: Buchausleihe

Rückgabe

Wird ein Buch zurückgegeben, wird zuvor auf den entsprechenden Button im Navigator geklickt. Anschließend erscheint ein kleines Fenster, das nach der Buchnummer fragt. Wird die richtige Buchnummer eingeben, erscheinen, wie die untere Abbildung zeigt, Rückgabeinformationen.
In diesem Rückgabeformular können keine Felder geändert werden. Außerdem sind nur zwei grafische Buttons, auch Icons genannt, auf dem Bildschirm vorhanden. Zum einen das Tür-Icon wie bei den anderen Formularen, zum anderen folgendes Rückgabe-Icon



Nach anklicken dieses Icons wird ein ausgeliehenes Buch in der Tabelle Ausleihe als zurückgegeben markiert und es wird zum Navigator zurückgesprungen. Des weiteren wird die Ausweisnummer des Ausleihers auf den Wert Null (0) gesetzt, da es wegen Datenschutzgründen nicht zulässig ist, einen Schüler auf Dauer zu speichern. Eine spätere Statistik ist durch diesen Vorgang nach wie vor möglich, da der Rest der Informationen erhalten bleibt.



Abbildung 6: Buchrückgabe

Abfragen...

Um mehrere Informationen auf einen Blick zu erhalten gibt es Abfragen. Diese werden über die Schaltfläche *Abfragen...* im Navigator aufgerufen. Darauf hin erscheint ein Untermenü mit den Auswahlmöglichkeiten

- Sachgebietssuche — sucht nach einem Sachgebiet
- Buchtitel suchen — sucht einen Buchtitel
- Autorensuche — sucht einen Autor
- Ausleiher suchen — sucht nach einem Schüler, der noch Bücher ausgeliehen hat
- Ausleihe > 3 Monate — zeigt alle Schüler an, die ihr Buch mehr als 3 Monate besitzen
- Buchnummer suchen — zeigt Informationen über eine Buchnummer an
- Zurück zum Hauptmenü — springt zurück zum Navigator

Abbildung 7: Abfrage-Untermenü

In diesem Untermenü können spezifische Informationen zu den einzelnen Gebieten gesucht und angezeigt werden.
Wird eine bestimmte Schaltfläche angeklickt, erscheint eine Abfragebox, die Eingaben zum jeweiligen Gebiet erwartet.

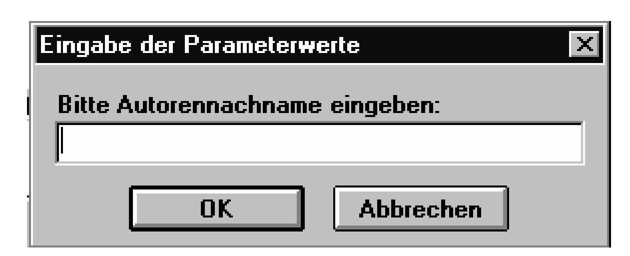


Abbildung 8 Abfragebox

Nach erfolgter Eingabe werden in einer Tabelle alle gefunden Einträge aufgelistet. Ist die Tabelle leer, wurden keine Angaben gefunden.

Bei der Texteingabe braucht die Klein-/Großschreibung nicht beachtet werden. Des weiteren sind Stellvertreterzeichen (*,?) zulässig. Ein ? steht für die Ersetzung eines beliebigen Buchstabens. So würden bei der Eingabe "Me?er" z.B. folgende Einträge gefunden: "Meier", "Meyer", "Meter", usw.

Ein Stern hingegen steht nicht für einen Buchstaben sondern für mehrere. So würden z.B. bei der Eingabe "Me*" alle Einträge gefunden, die mit "Me" beginnen.

Beenden

Nach betätigen dieser Schaltfläche wird die Bibliotheksverwaltung nach einer Abfrage verlassen. Man befindet sich nun wieder im Programmanager.

Fehlermeldungen

2

Es konnte kein neuer Datensatz angelegt werden!

Es wurde versucht einen Datensatz zu erstellen, der

1. unvollständig ausgefüllt ist
2. gegen die referentielle Integrität verstößt, sprich doppelt vorkommen würde.

Problemlösung:

- eine andere/n Wert vergeben, der nicht doppelt vorkommt
- das Formular vollständig ausfüllen
- auf das Icon Radierer (=Rückgängig) klicken

Kann <*Formularname*> nicht verlassen!

Es wurde versucht einen neuen Datensatz anzulegen, der

1. unvollständig ausgefüllt ist
2. gegen die referentielle Integrität verstößt, sprich doppelt vorkommen würde.

Problemlösung:

- eine andere/n Wert vergeben, der nicht doppelt vorkommt
- das Formular vollständig ausfüllen
- auf das Icon Radierer (=Rückgängig) klicken

Datensatz wurde nicht gelöscht!

Es wurde versucht

1. eine unvollständig oder leere Eingabemaske zu löschen

Problemlösung:

- auf das Icon Radierer (=Rückgängig) klicken

<Datensatz rückgängig> konnte nicht ausgeführt werden!

Es wurde versucht bei einer leere Eingabemaske rückgängig zu machen.

Problemlösung:

- keine - eine leere Eingabemaske kann nicht rückgängig gemacht werden!

Das eingegebene Buch wird bereits ausgeliehen!

Es wurde versucht ein Buch, das bereits verliehen ist, nochmals auszuleihen.

Problemlösung:

- es wurde vergessen das Buch bei der Rückgabe dementsprechend zu markieren; dies muß dann nachgeholt werden
- bei der manuellen Verteilung der Buchnummer wurde diese versehentlich doppelt vergeben; dies ist zu korrigieren

Es existiert kein Schüler der sein Buch zurückgeben könnte!

Bei der Rückgabe wurde im Abfragefenster eine falsche Nummer eingegeben. Dies wird dadurch ersichtlich, daß keine Informationen im Formular Rückgabe angezeigt werden.

Problemlösung:

- Buchrückgabe beenden und nochmals aufrufen - diesmal die richtige Nummer eingeben.

Literaturverzeichnis

Burkert; Lächa: *Datenbanken – Informatik für die Sekundarstufe II*, 2. Auflage, Hessisches Landesinstitut für Pädagogik (HeLP), 2002 [Vorlage für vorliegende Überarbeitung]

Fischer et al.: *Grundlagen der Informatik II*, Oldenbourg-Verlag, 2007-09-19

Kemper, Alfons: *Datenbanksysteme – Eine Einführung*, Universität Passau, 2007

Kowalski, Michael: *Relationale Datenbanken*, Lehrer-Online, 2005

Noack, Wilhelm [Hrsg.]: *SQL – Grundlagen und Datenbankdesign*, 4.Auflage, Herdt-Verlag, Ausgabe für das RRZN-Niedersachsen/Hannover, 2005

Steuber, Tanja: *Microsoft Office – Access 2003, Grundlagen für Datenbankentwickler*, Herdt-Verlag, 2006

Vornberger: *Vorlesung Datenbanken*, Universität Osnabrück, FB Informatik, 2007